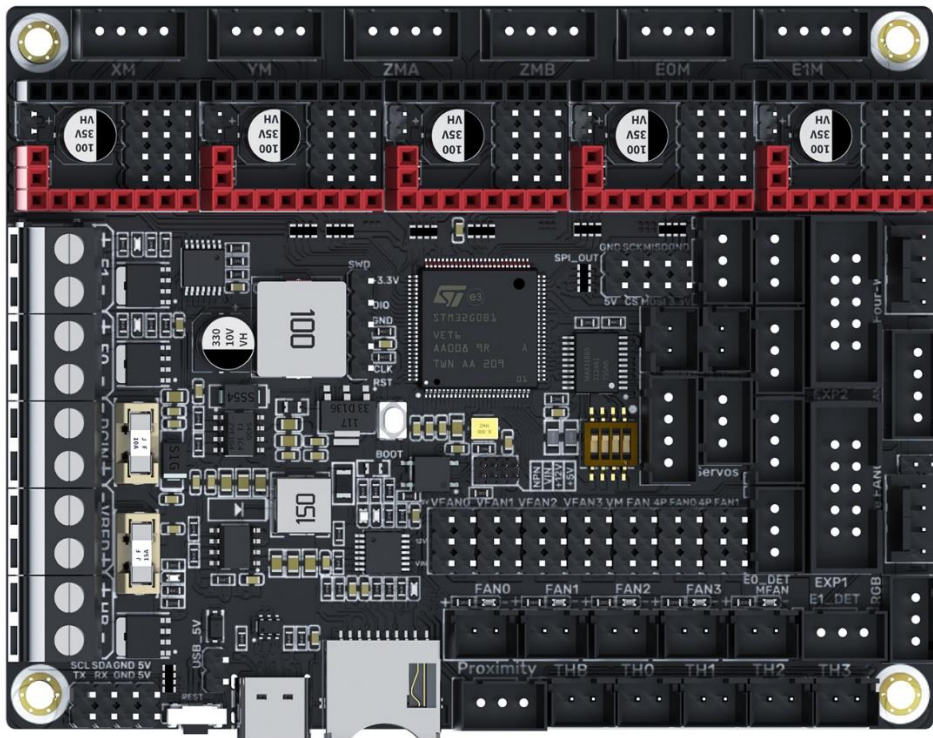


BIGTREETECH

SKRat v1.0

User Manual



Contents

| | |
|---|----|
| Revision History | 5 |
| Product Profile | 6 |
| Feature Highlights | 6 |
| Specifications | 7 |
| Firmware | 7 |
| Dimensions | 8 |
| Peripheral Port | 9 |
| Connector Diagram | 9 |
| Pinout Diagram | 9 |
| Function Introduction | 10 |
| LED Indicator Light | 10 |
| USB Power Supply | 10 |
| Downloading Firmware via DFU | 11 |
| To Choose the Working Mode of the TMC Driver | 11 |
| STEP/DIR | 11 |
| UART | 12 |
| SPI | 13 |
| Sensorless Homing | 13 |
| Fan Voltage Selection | 14 |
| +5V | 14 |
| 12V | 14 |
| VIN | 14 |
| 2-pin Fan Wiring | 15 |
| 4-pin Fan Port Wiring | 15 |
| Proximity Switch | 16 |
| Voltage Selection | 16 |
| PNP/NPN Type | 16 |

Shenzhen Big Tree Technology Co., Ltd.
BIGTREETECH

| | |
|---|----|
| Proximity Switch Wiring | 17 |
| Auto Power off (BIGTREETECH Relay V1.2) Wiring | 17 |
| Power Loss Recovery (BTT UPS 24V V1.0) Wiring | 18 |
| RGB Wiring | 18 |
| BLTouch Wiring | 19 |
| Filament Sensor Wiring | 19 |
| LCD Display Wiring | 20 |
| SPI, I2C, UART | 20 |
| 100K NTC | 21 |
| CAN bus Wiring | 22 |
| Marlin | 23 |
| Install Compiling Environment | 23 |
| Download Marlin Firmware | 23 |
| Configure Firmware | 23 |
| Open Marlin Project | 23 |
| Compiling Environment | 23 |
| Configure Motherboard and Serial Port | 24 |
| Configure Stepper Driver | 25 |
| Sensorless Homing | 25 |
| 100K NTC | 26 |
| Max31865 | 26 |
| BLTouch | 27 |
| Auto Power Off (Relay V1.2) | 30 |
| Power Loss Recovery | 30 |
| RGB | 31 |
| Filament Sensor | 32 |
| Smart Filament Sensor(SFS V1.0) | 32 |
| Compile Firmware | 34 |

Shenzhen Big Tree Technology Co., Ltd.
BIGTREETECH

| | |
|---|----|
| Klipper | 35 |
| Using Raspberry Pi | 35 |
| Download OS Image | 35 |
| Download and Install Raspberry Pi Imager | 35 |
| Write Image | 36 |
| Using BIGTREETECH CB1 | 39 |
| Download OS Image | 39 |
| Download and Install balenaEtcher | 39 |
| Write OS | 39 |
| WiFi Setting | 41 |
| Configure the Motherboard | 42 |
| SSH Connect to Device | 42 |
| Compile MCU Firmware | 43 |
| Firmware Update | 44 |
| Configure Klipper | 46 |
| Firmware Update | 48 |
| Cautions | 48 |
| FAQ | 48 |

Revision History

| Version | Note | Date |
|---------|----------|------------|
| 01.00 | Original | 2023/02/25 |

Product Profile

BIGTREE TECH SKRat v1.0 is a 3D printer motherboard jointly launched by Shenzhen Big Tree Technology Co., Ltd. and Rat Rig.

Feature Highlights

- 32-bit 64 MHz ARM Cortex-M0+ series STM32G0B1VET6 MCU.
- Onboard BOOT button to enable DFU mode to update the bootloader.
- The thermistor circuit is protected to prevent MCU damage from shorted heated bed and heater cartridge connection.
- Four 2-pin fan ports, one always-on fan port, and two 4-pin fan ports.
- All fans can realize VIN, 12V, 5V voltage selection via jumpers, and different voltages can be set separately for different ports.
- Integrated SPI and UART mode of TMC driver and DIAG pin, easily configurable with jumpers.
- Supports power loss recovery, filament runout sensor, auto power-off, BLTouch, proximity switch, RGB, etc.
- Onboard non-spring loaded microSD card slot for upgrading and configuring firmware through microSD card, which is simple, convenient, and efficient.
- Onboard CAN bus port.
- The SPI expansion port is +3.3V and +5V selectable, which is convenient to connect expansion modules, such as an ADXL345 accelerometer.
- Onboard UART and I2C expansion output ports.

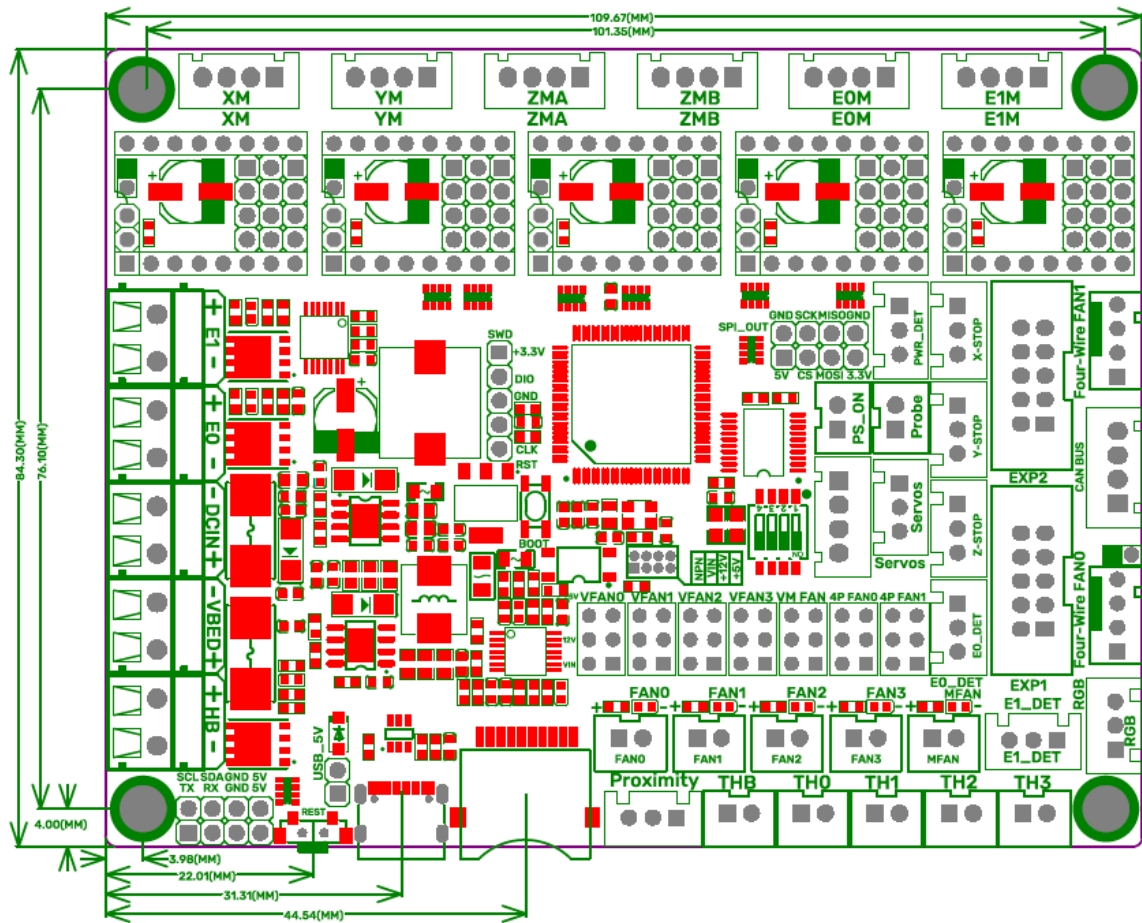
Specifications

| | |
|--|---|
| Dimensions | 110 x 85mm, for details please refer to BTT SKRat_V1.0_SIZE |
| Mounting Size | 110 x 85mm |
| MCU | ARM Cortex-M0+ STM32G0B1VET6 |
| Input Voltage | DC12V-DC24V |
| Logic Voltage | DC 3.3V |
| Heater Connection | Heated Bed (HB), Heater Cartridge (HE0, HE1) |
| HB Port Max. Current | 10A Continuous, 15A Instantaneous |
| Heater Cartridge Max. Current | 8A Continuous, 10A Instantaneous |
| Fan Port | 4 x CNC, 1 x Always On, 2 x 4-pin Header |
| Fan Port Max. Current | 1A Continuous, 1.5A Instantaneous |
| Overall Current (Heater Cartridge +Drivers+All Fans) | <15A |
| Expansion Port | BLTouch (Servos, Probe), PS-ON, PWR-DET, Fil-DET, RGB, CAN-FD, SPI, UART, I2C |
| Stepper Driver Mode | SPI, UART, STEP/DIR |
| Stepper Motor Socket | X, Y, Z (Dual Z axes), E0, E1 5 channels in total |
| Thermistor | 5 x NTC Ports, 1 x PT100/PT1000 |
| Display | 2.4-inch TFT, 3.5-inch TFT, LCD12864 Screen... |
| PC Connection | Type-C |
| Supported File Format | G-code |
| Recommended Slicer/Console | Cura, Simplify3D, Pronterface, Repetier-host, Makerware |

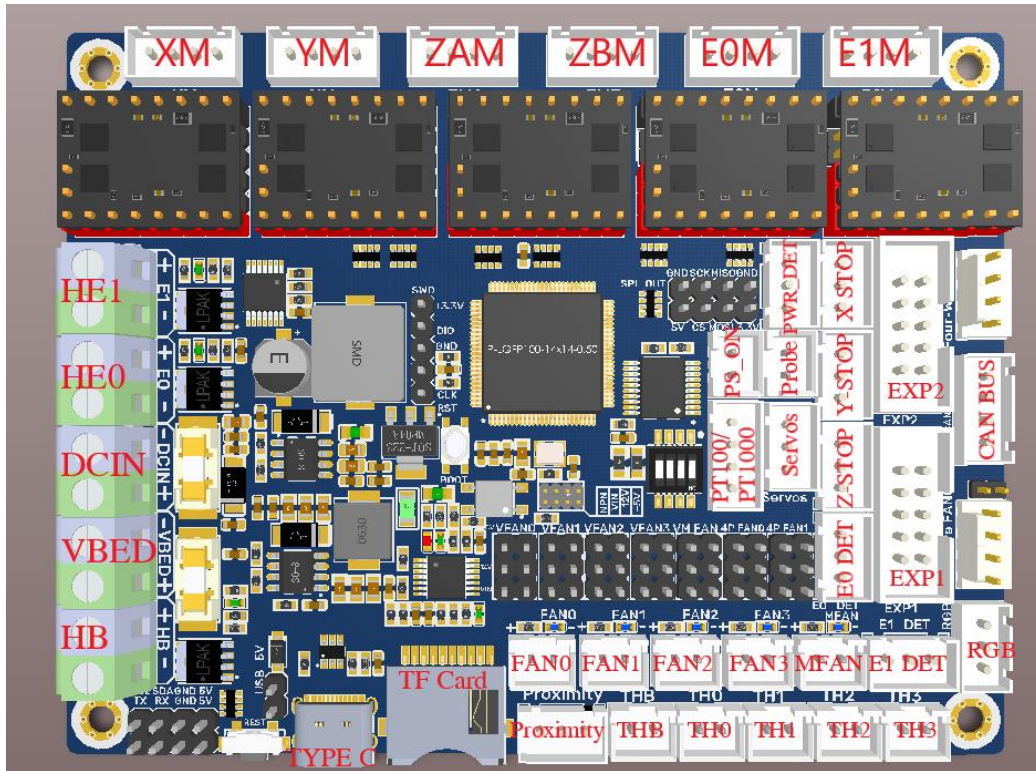
Firmware

Supported Firmware: Marlin, Klipper

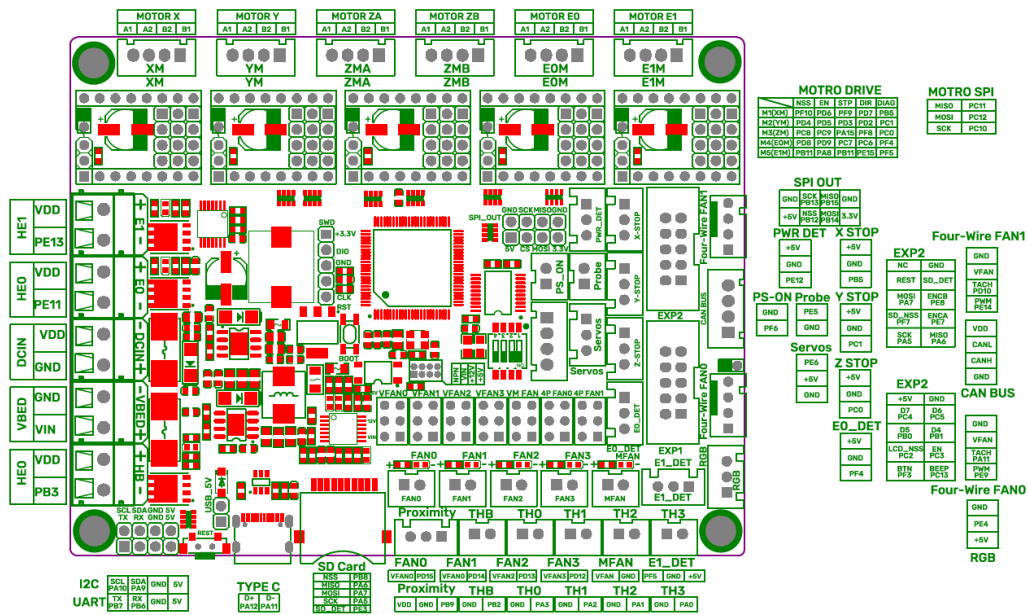
Dimensions



Peripheral Port Connector Diagram



Pinout Diagram



For details please refer to BTT SKRat_V1.0_SIZE.

Function Introduction

LED Indicator Light

After the motherboard is powered on:

Power-Red Light-Power Indicator: The solid red light indicates normal motherboard power.

Status-Green Light-Status Indicator: When updating firmware, this light will flash and then be controlled by the firmware.

D10-Green Light-HB (Heated Bed) Status Indicator: The light will remain solid green when the heated bed is working and turn off when it is not.

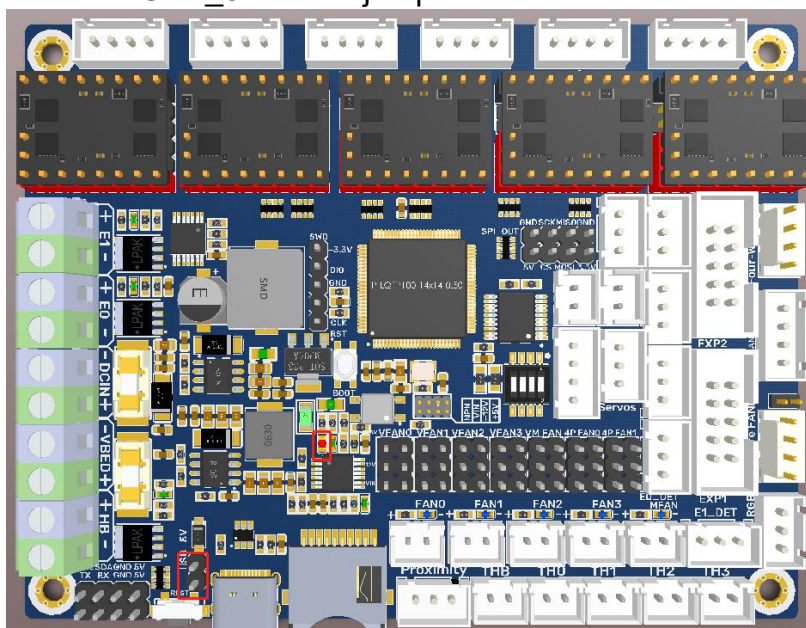
D3, D6-Green Light-E0, E1 (Heater Cartridge) Status Indicator: The light will remain solid green when the heater cartridge is working and turn off when it is not.

FAN0, FAN1, FAN2, FAN3-Blue Light-CNC Fan Status Indicators: The blue lights turn on when the corresponding CNC fan is running and turn off when the fan is off.

MFAN-Blue Light-MFAN Status Indicator: The blue light will remain on when the power supply is normal.

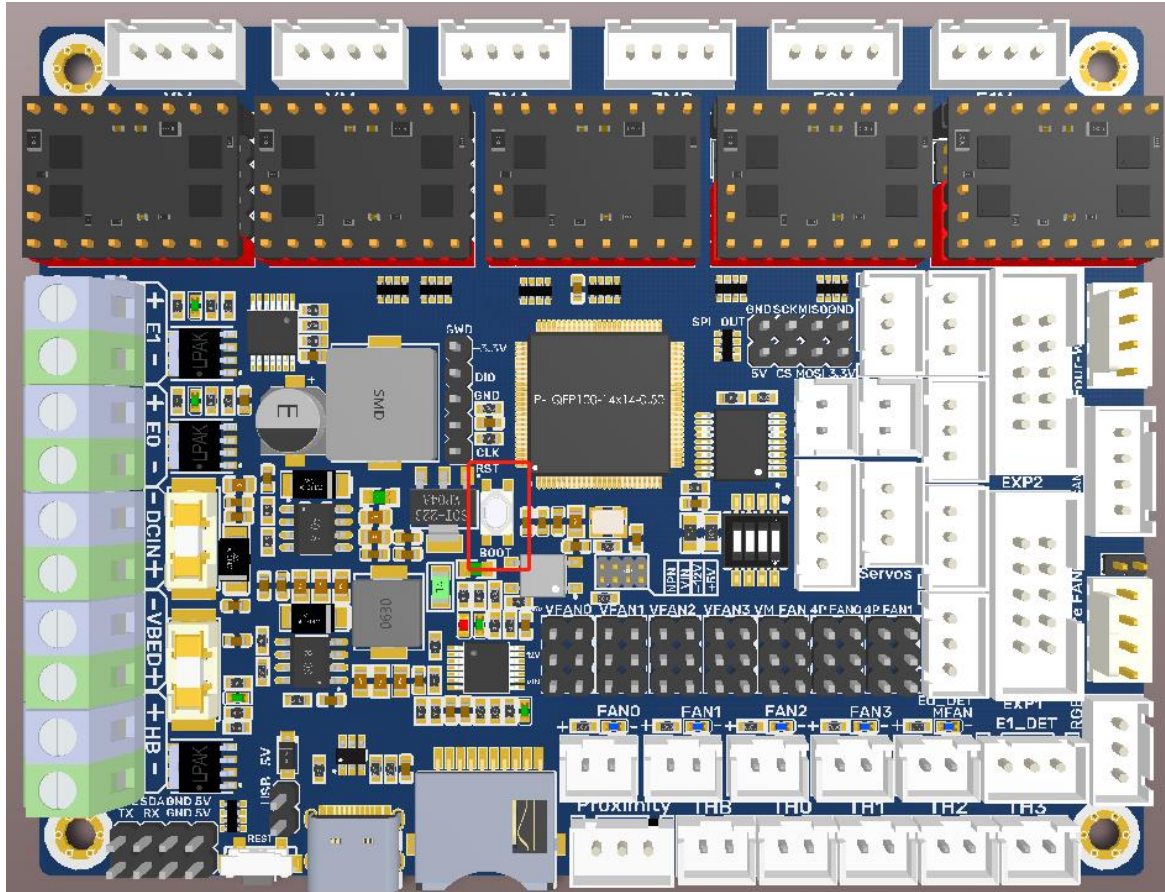
USB Power Supply

After the SKRat has been powered, the Power-Red Light on the middle of the board will light up, indicating power on. If using USB to power the board, please short the USB_5V with a jumper.



Downloading Firmware via DFU

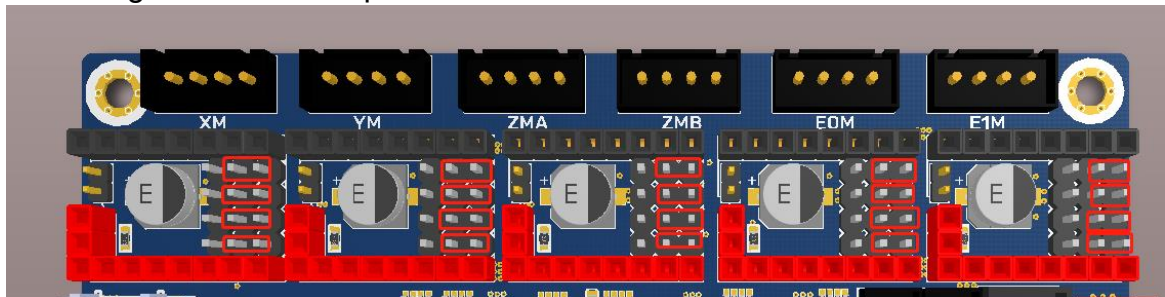
Press and hold the BOOT button, power on the motherboard, and the chip will enter the DFU mode. At this time, you can connect the board to the PC via the Type-C port, and update the firmware via DFU mode.



To Choose the Working Mode of the TMC Driver

STEP/DIR

e.g.: A4988, DRV8825, LV8729, ST820, etc., connect jumpers(MS0-MS2) according to the microstep table below.



| Driver Chips | MS1 | MS2 | MS3 | Microsteps | Excitation Mode |
|--|---|-----|-----|------------|-----------------|
| A4988 Max. 16 Microsteps 35V 2A | L | L | L | Full Step | 2 Phase |
| | H | L | L | 1/2 | 1-2 Phase |
| | L | H | L | 1/4 | W1-2 Phase |
| | H | H | L | 1/8 | 2W1-2 Phase |
| | H | H | H | 1/16 | 4W1-2 Phase |
| Current $R_S=0.1\Omega$ | $I_{\text{TripMAX}} = \frac{V_{\text{REF}}}{8 * R_S}$ | | | | |

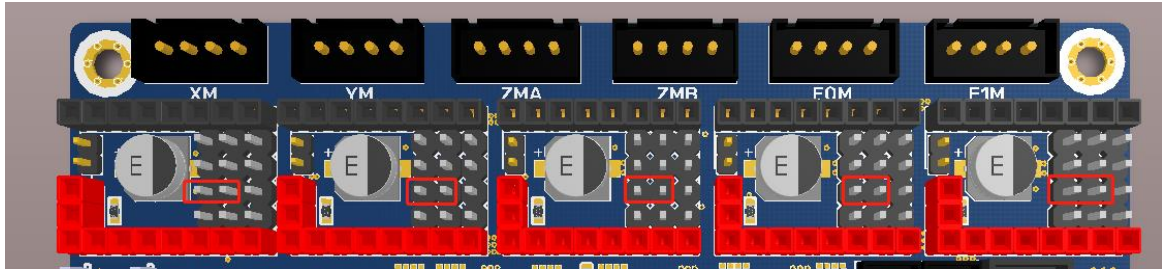
| Driver Chips | MODE2 | MODE1 | MODE0 | Microsteps | Excitation Mode |
|--|---|-------|-------|------------|-----------------|
| DRV8825 Max. 32 Microsteps 8.2V-45V 2.5A at 24V T=25°C | L | L | L | Full Step | 2 Phase |
| | L | L | H | 1/2 | 1-2 Phase |
| | L | H | L | 1/4 | W1-2 Phase |
| | L | H | H | 1/8 | |
| | H | L | L | 1/16 | |
| | H | L | H | 1/32 | |
| | H | H | L | 1/32 | |
| | H | H | H | 1/32 | |
| Current $R_{\text{ISENSE}}=0.1\Omega$ | $I_{\text{CHOP}} = \frac{V_{(\text{xREF})}}{5 * R_{\text{ISENSE}}}$ | | | | |

| Driver Chips | MD3 | MD2 | MD1 | Microsteps | Excitation Mode |
|--|---|-----|-----|------------|-----------------|
| LV8729 Max. 128 Microsteps 36V 1.8A | L | L | L | Full Step | 2 Phase |
| | L | L | H | 1/2 | 1-2 Phase |
| | L | H | L | 1/4 | W1-2 Phase |
| | L | H | H | 1/8 | 2W1-2 Phase |
| | H | L | L | 1/16 | 4W1-2 Phase |
| | H | L | H | 1/32 | 8W1-2 Phase |
| | H | H | L | 1/64 | 16W1-2 Phase |
| | H | H | H | 1/128 | 32W1-2 Phase |
| Current $RF1=0.22\Omega$ | $I_{\text{OUT}} = (V_{\text{REF}} / 5) / RF1$ | | | | |

| Driver Chips | MS3 | MS2 | MS1 | Microsteps |
|---|--|-----|-----|------------|
| ST820 Max. 256 Microsteps 45V 1.5A | L | L | L | Full Step |
| | L | L | H | 1/2 |
| | L | H | L | 1/4 |
| | L | H | H | 1/8 |
| | H | L | L | 1/16 |
| | H | L | H | 1/32 |
| | H | H | L | 1/128 |
| | H | H | H | 1/256 |
| Current $R_S=0.15\Omega$ | $I_{\text{peak}} = \frac{V_{\text{REF}} * V_{\text{DD}}}{5 * R_S}$ | | | |

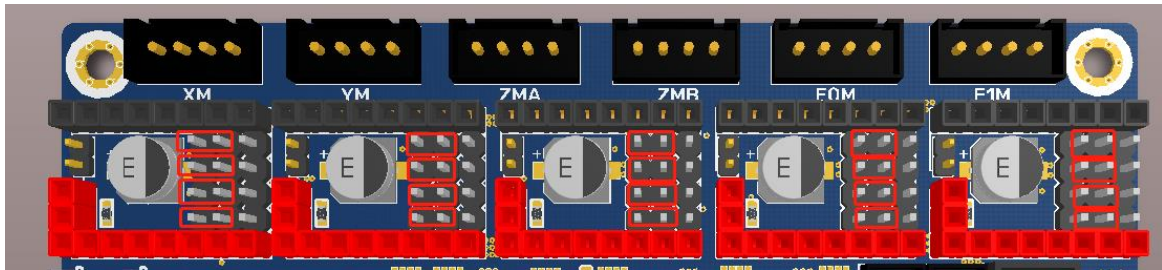
UART

e.g.: TMC2208, TMC2209, TMC2225, etc., place jumpers according to the diagram below, microstep and current can be configured in firmware.



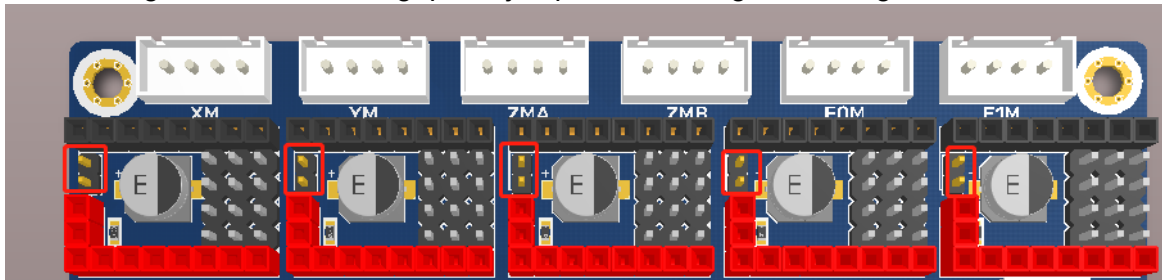
SPI

e.g.: TMC2130, TMC5160, TMC5161, etc., place jumpers according to the diagram below, microstep and current can be configured in firmware.



Sensorless Homing

When using sensorless homing, place jumpers according to the diagram below.

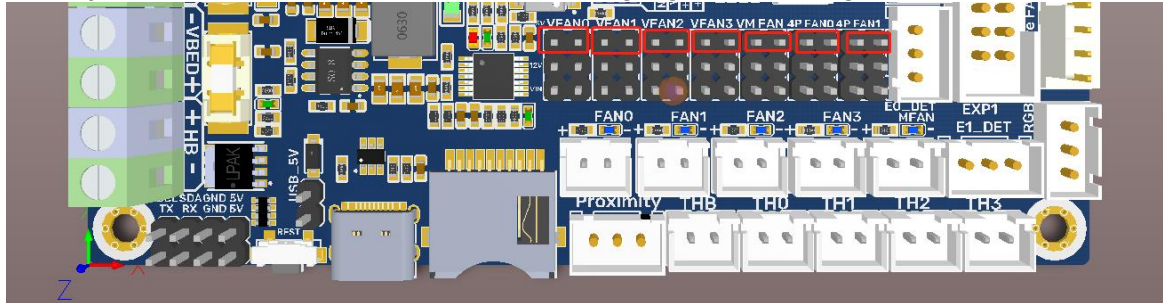


Note: The extra ENDSTOP cannot be used by selecting this function!!!

Fan Voltage Selection

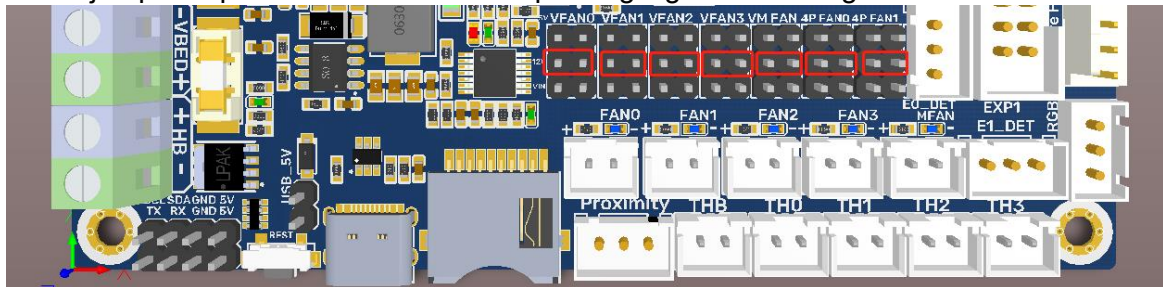
+5V

Use a jumper cap to short the horizontal pins highlighted in the figure below.



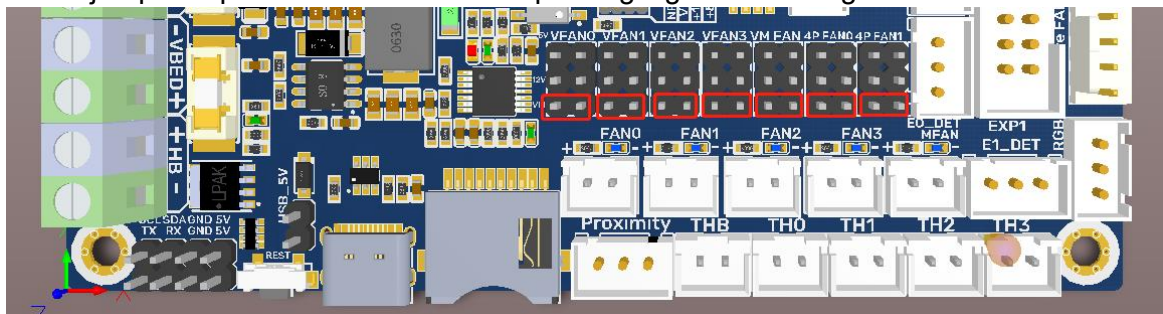
12V

Use a jumper cap to short the horizontal pins highlighted in the figure below.

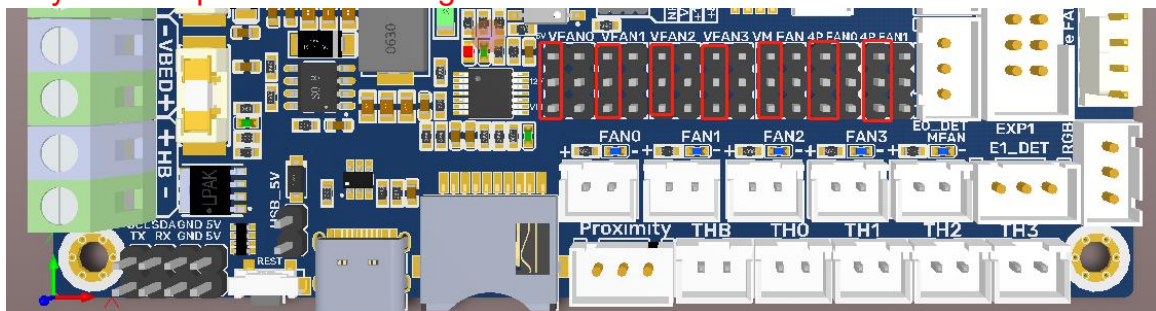


VIN

Use a jumper cap to short the horizontal pins highlighted in the figure below.



Note: Shorting the pins, as shown in the figure below, is strictly prohibited as it may result in permanent damage to the motherboard.



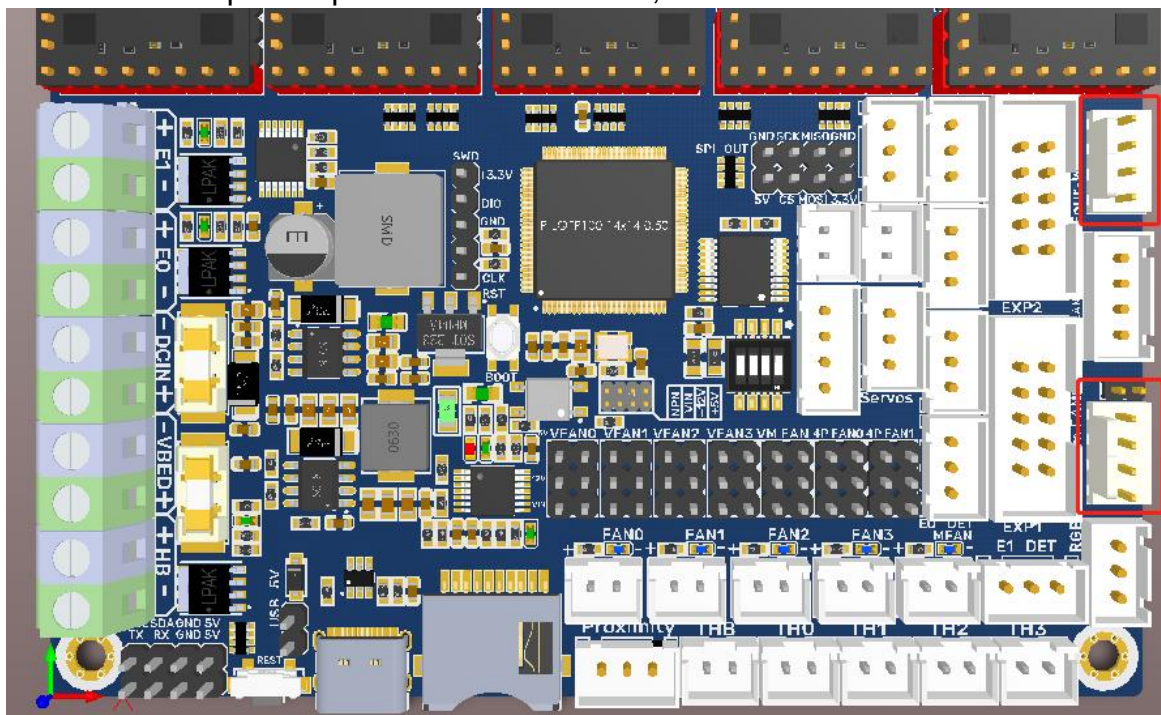
2-pin Fan Wiring

Onboard four 2-pin fan ports (FAN0,FAN1,FAN2,FAN3), one always on fan port(MFAN).



4-pin Fan Port Wiring

Onboard two 4-pin fan ports: Four-Wire FAN0, Four-Wire FAN1

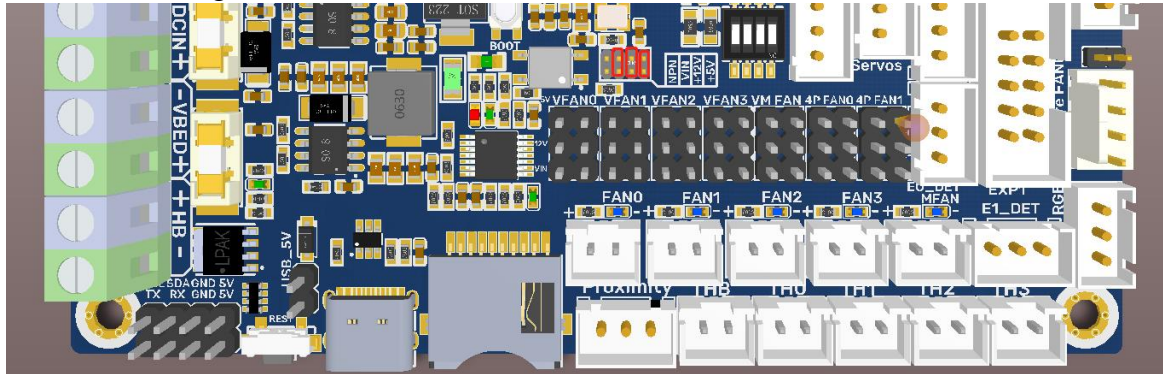


Note: Using a voltage that does not match a fan's rated working voltage can cause abnormal fan operation or damage. Always ensure that the selected voltage matches the fan's rated working voltage.

Proximity Switch

Voltage Selection

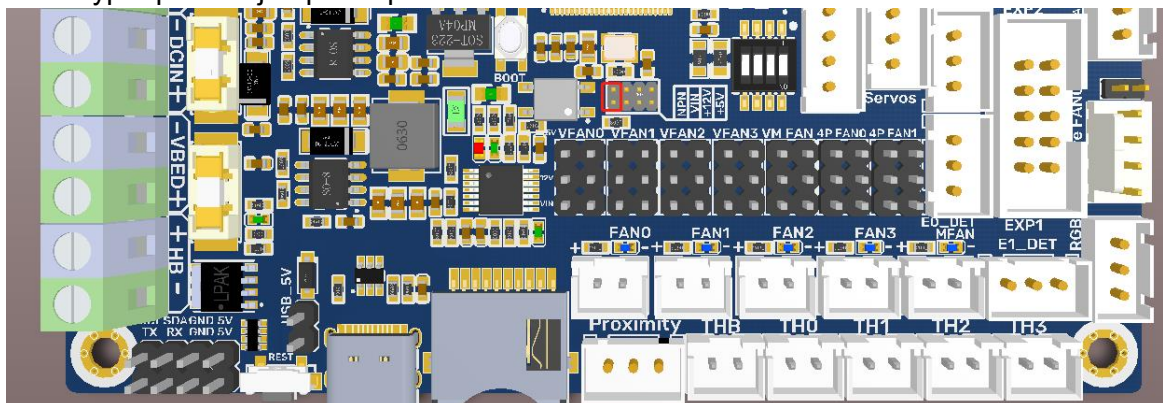
Use a jumper cap to short the corresponding pin to choose +12V, +5V, VIN, as shown in the figure below.



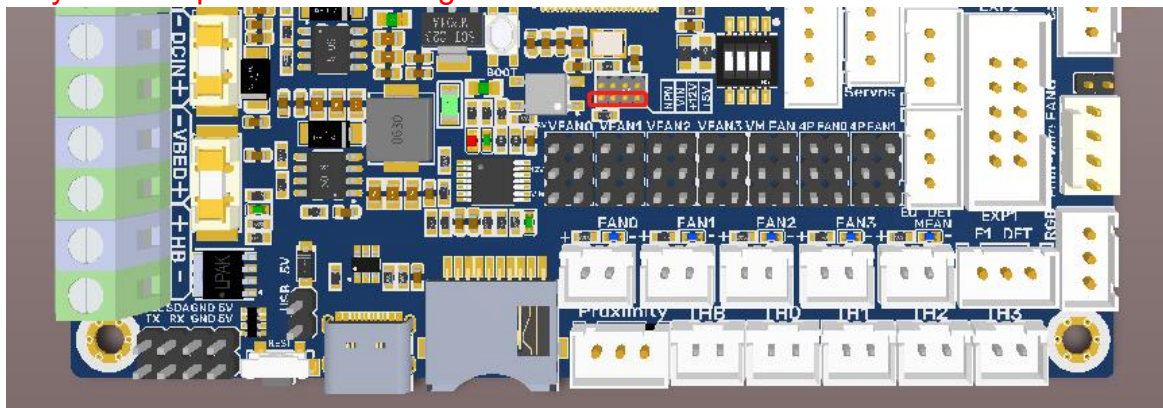
PNP/NPN Type

PNP Type: no need for a jumper cap.

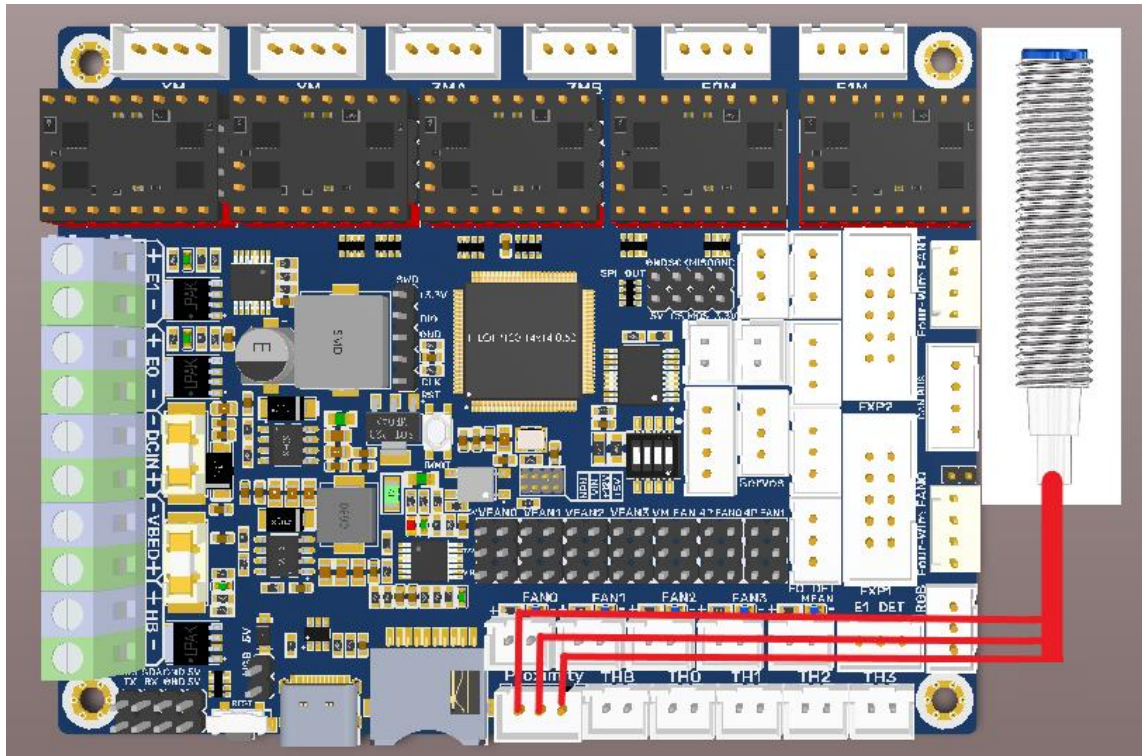
NPN Type: place a jumper cap.



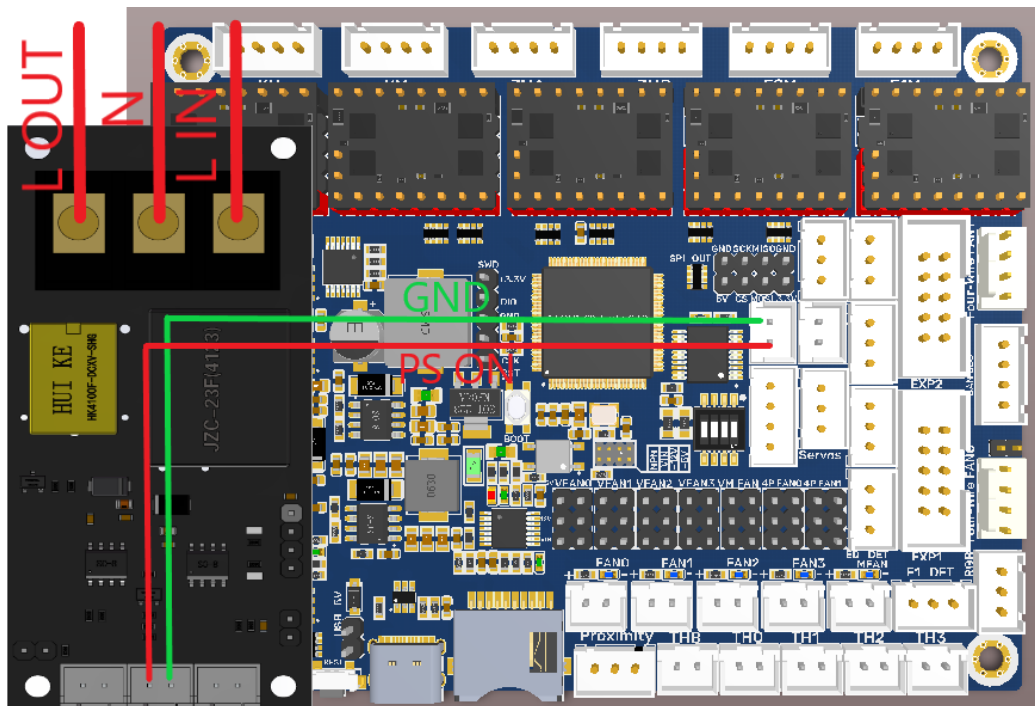
Note: Shorting the pins, as shown in the figure below, is strictly prohibited as it may result in permanent damage to the motherboard.



Proximity Switch Wiring



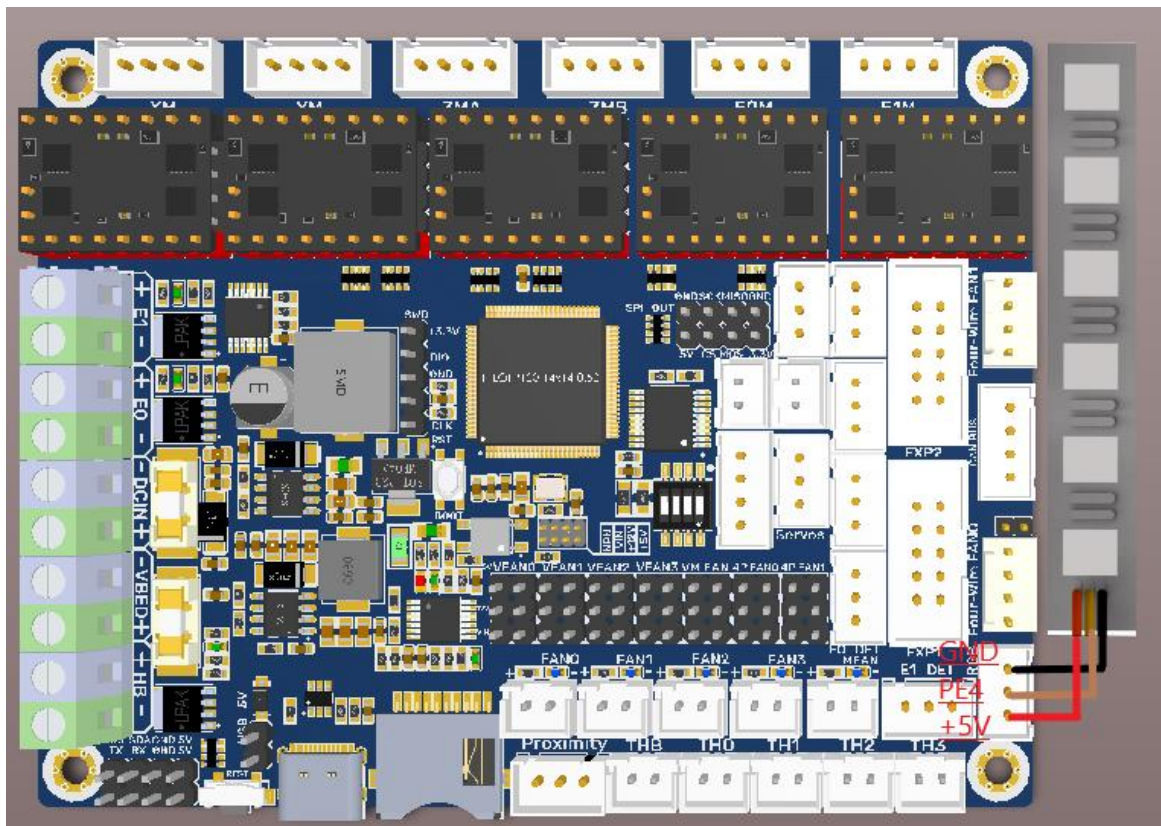
Auto Power off (BIGTREETECH Relay V1.2) Wiring



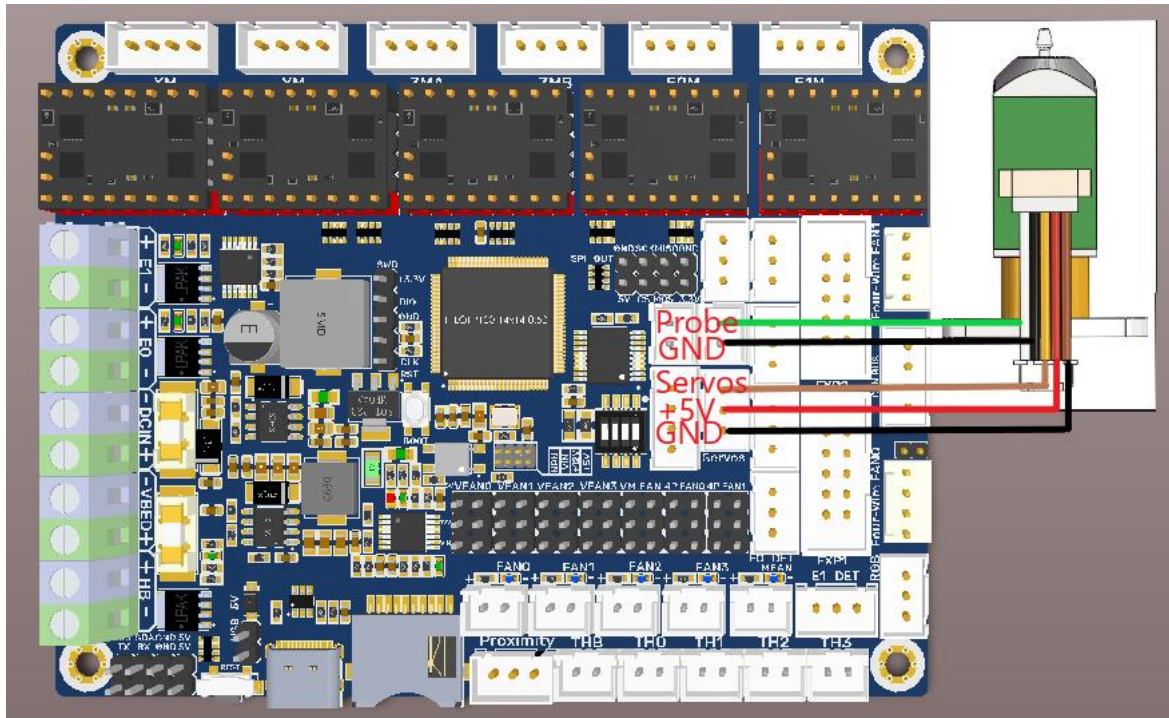
Power Loss Recovery (BTT UPS 24V V1.0) Wiring



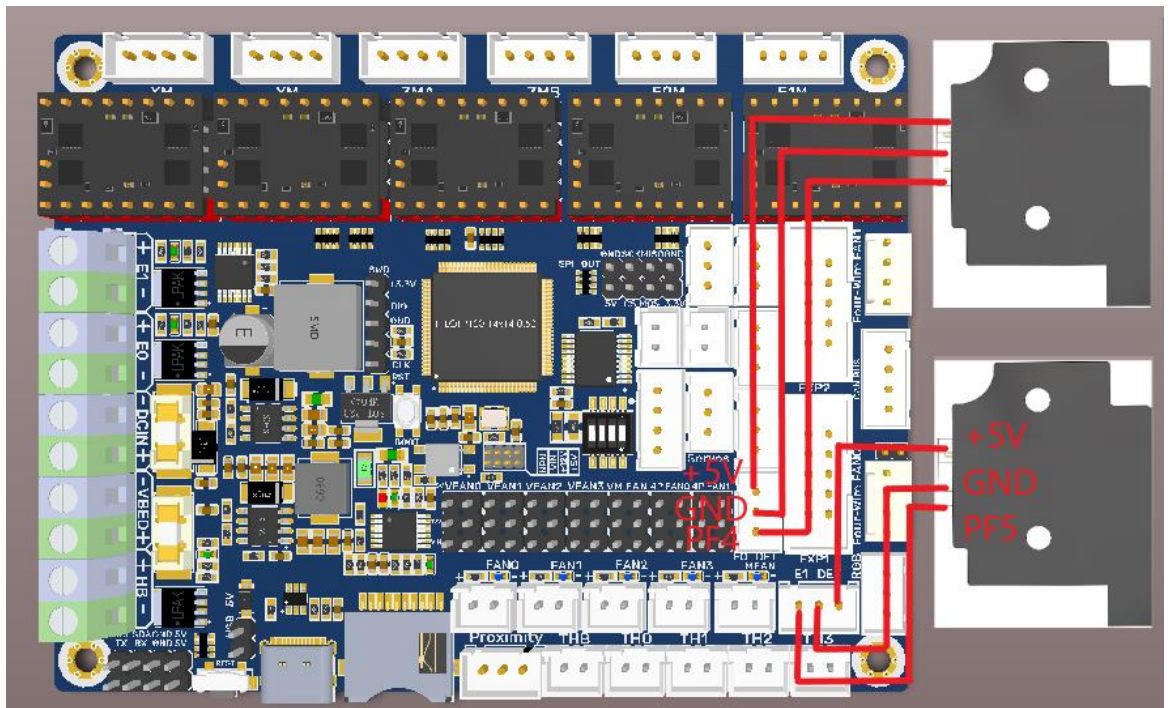
RGB Wiring



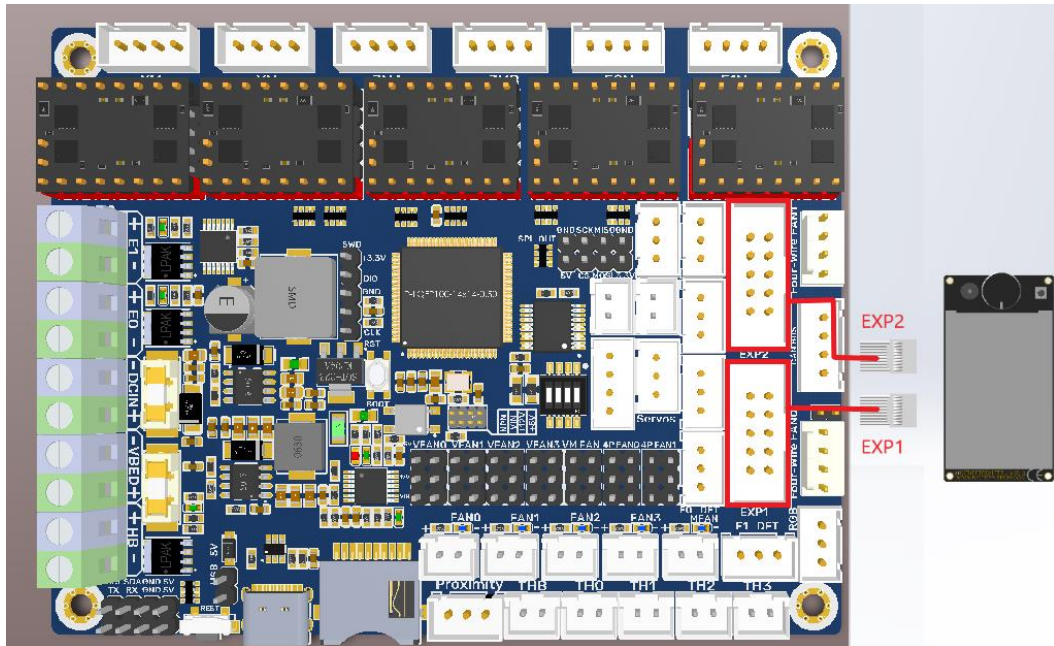
BLTouch Wiring



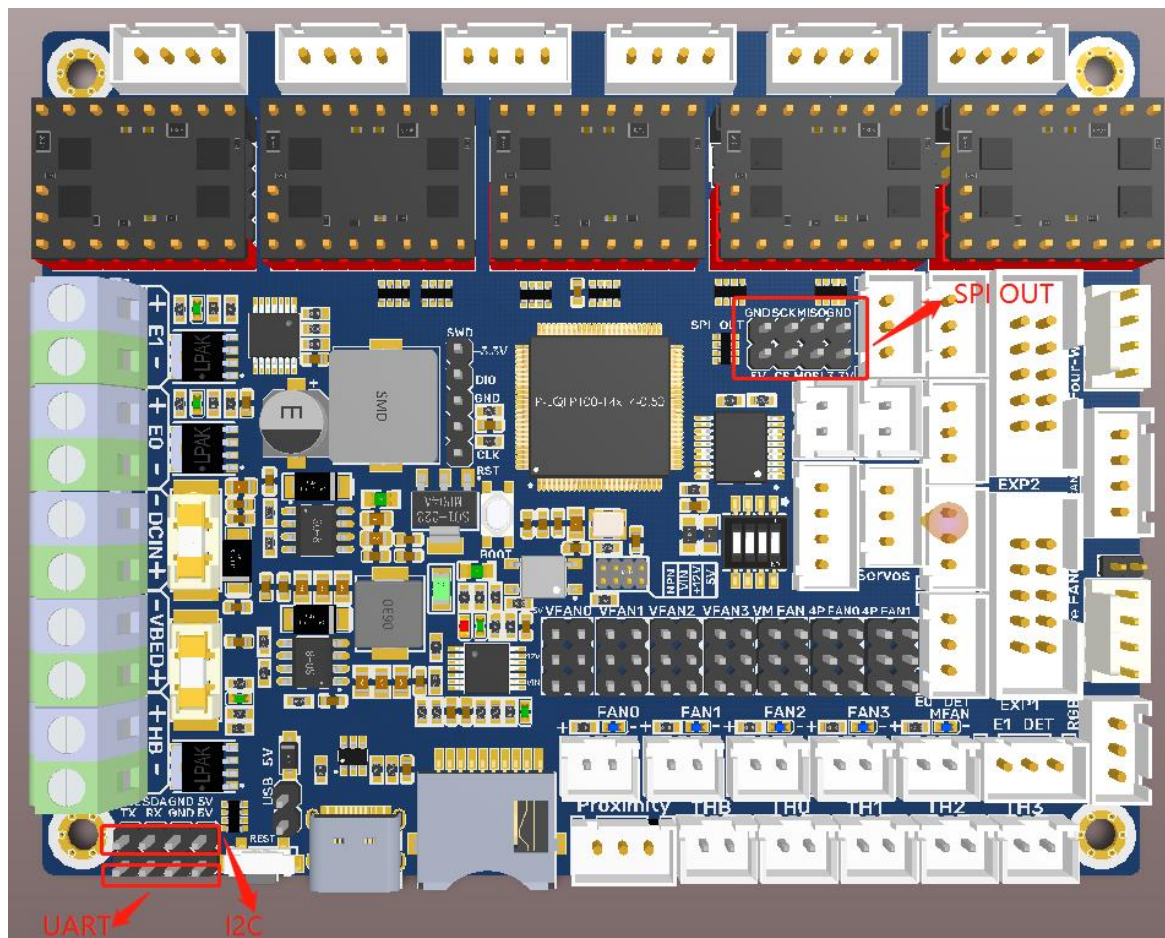
Filament Sensor Wiring



LCD Display Wiring

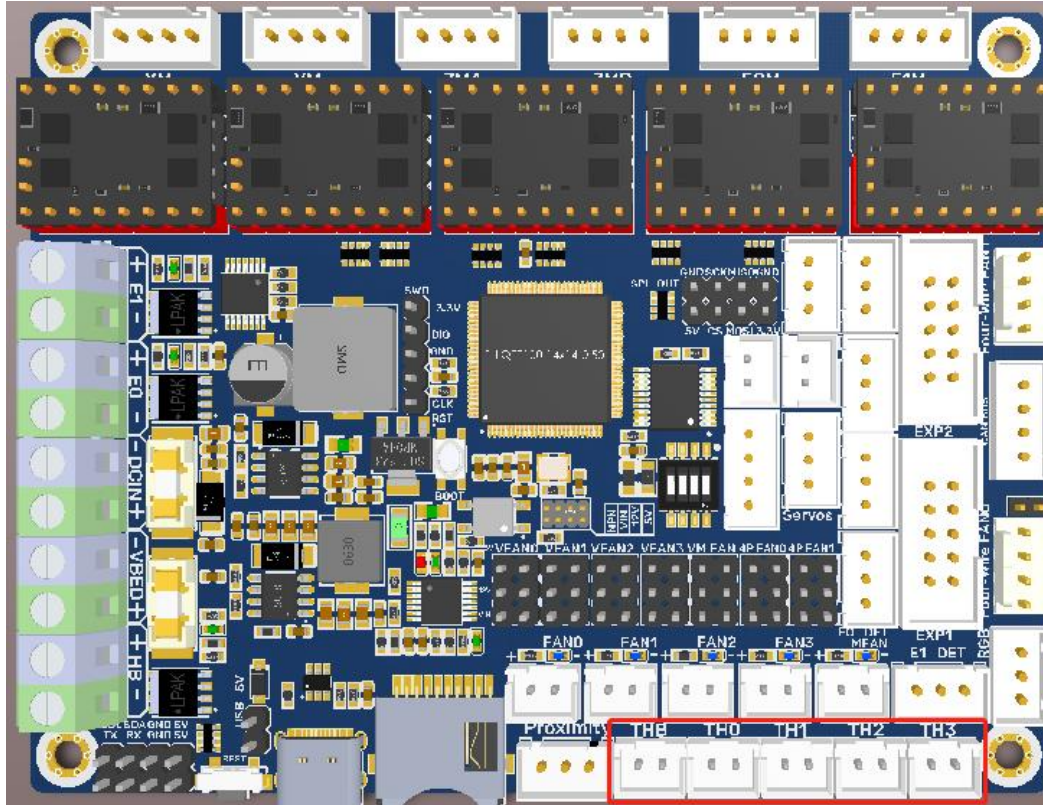


SPI, I2C, UART

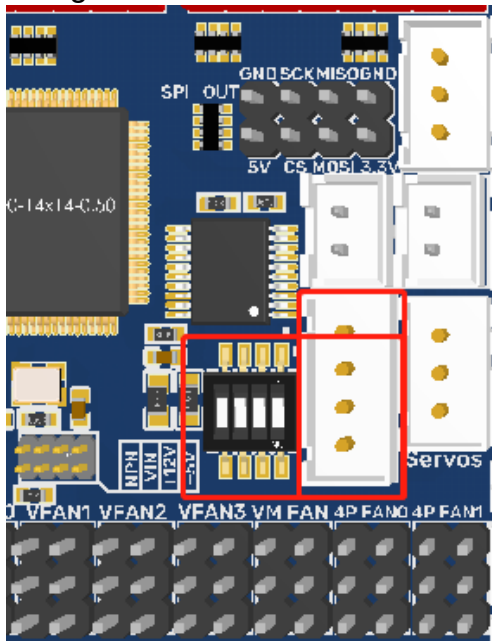


100K NTC

When using a 100K NTC thermistor, the NTC resistor is connected to THB, TH0-TH3, and the pull-up resistance of TH0-TH3 is 4.7K 0.1%.



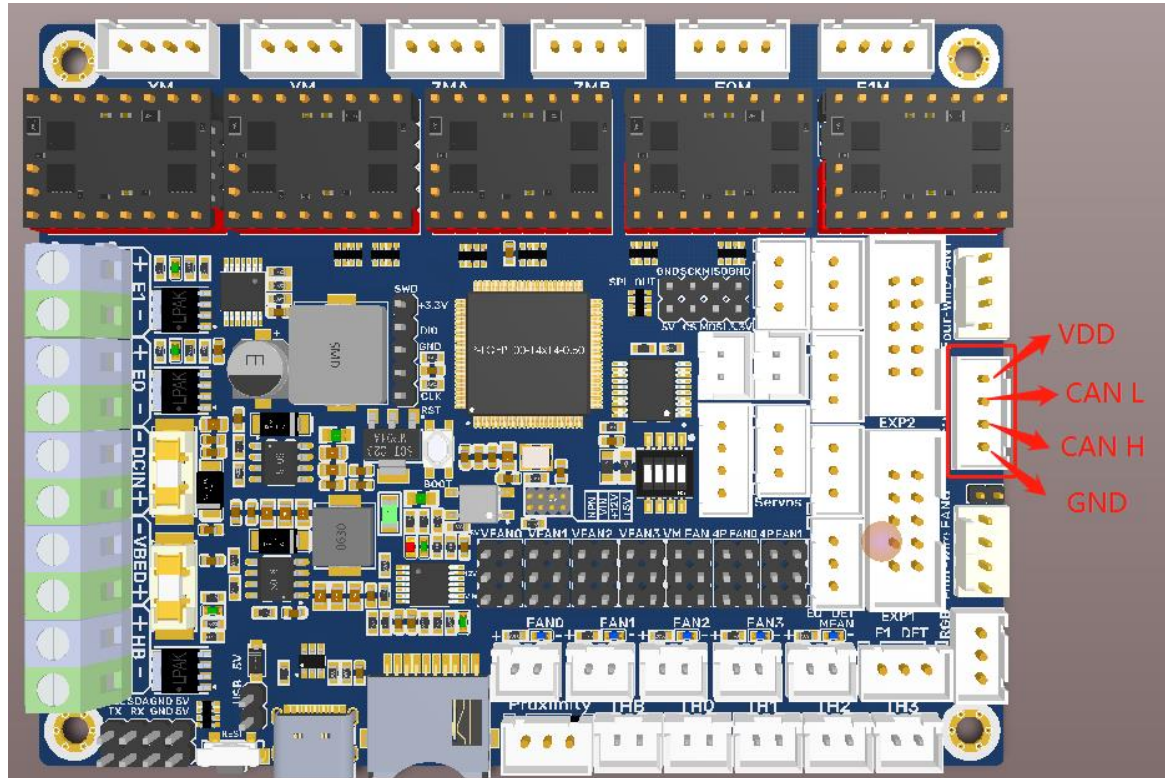
Using PT100/PT1000 via MAX31865:



| 1 | 2 | 3 | 4 | SensorModel |
|-----|-----|-----|-----|---------------|
| ON | ON | ON | OFF | 2lines PT00 |
| ON | ON | OFF | ON | 2lines PT1000 |
| OFF | OFF | ON | OFF | 4lines PT100 |
| OFF | OFF | OFF | ON | 4lines PT1000 |

CAN bus Wiring

Connect the CAN bus sensor to the CAN bus port on the board.



Note: The output voltage of the motherboard CAN bus port is the same as its input voltage.

Marlin

Install Compiling Environment

<https://github.com/bigtreetech/Document/blob/master/How%20to%20install%20VScode%2BPlatformio.md>
https://marlinfw.org/docs/basics/install_platformio_vscode.html

Refer to the link above for tutorial on installing VSCode and PlatformIO plugin.

Download Marlin Firmware

Download the newest bugfix version of Marlin from the official website:

<https://github.com/MarlinFirmware/Marlin/tree/bugfix-2.0.x>

Configure Firmware

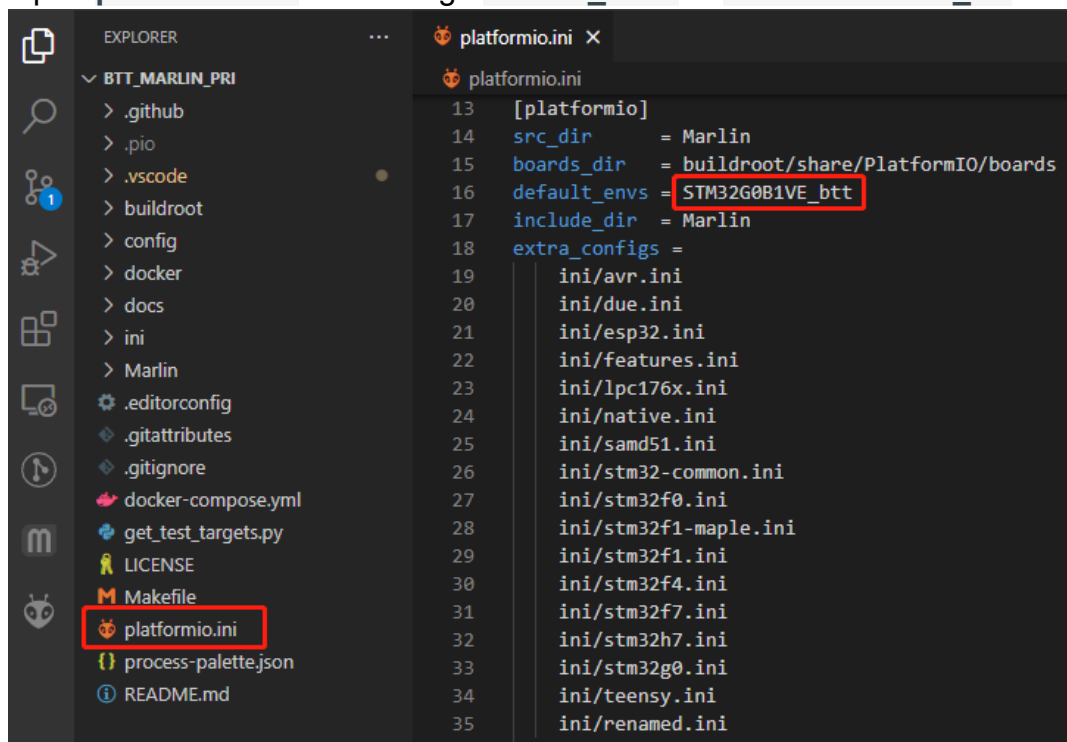
Open Marlin Project

You can open Marlin in VSCode in one of the following ways:

- Drag the downloaded Marlin Firmware folder onto the VSCode application icon.
- Use the **Open...** command in the VSCode **File** menu.
- Open the PIO Home tab and click the **Open Project** button.

Compiling Environment

Open **platformio.ini** and change **default_envs** to **STM32G0B1VE_btt**.



The screenshot shows the VS Code interface with the Explorer view on the left and the platformio.ini file open in the editor. The Explorer view shows the project structure with 'platformio.ini' highlighted. The editor shows the following content:

```
platformio.ini
13 [platformio]
14 src_dir = Marlin
15 boards_dir = buildroot/share/PlatformIO/boards
16 default_envs = STM32G0B1VE_btt
17 include_dir = Marlin
18 extra_configs =
19     ini/avr.ini
20     ini/duemiliani.ini
21     ini/esp32.ini
22     ini/features.ini
23     ini/lpc176x.ini
24     ini/native.ini
25     ini/samd51.ini
26     ini/stm32-common.ini
27     ini/stm32f0.ini
28     ini/stm32f1-maple.ini
29     ini/stm32f1.ini
30     ini/stm32f4.ini
31     ini/stm32f7.ini
32     ini/stm32h7.ini
33     ini/stm32g0.ini
34     ini/teensy.ini
35     ini/renamed.ini
```

Configure Motherboard and Serial Port

Set MOTHERBOARD to **BOARD_BTT_SKRAT_V1_0**

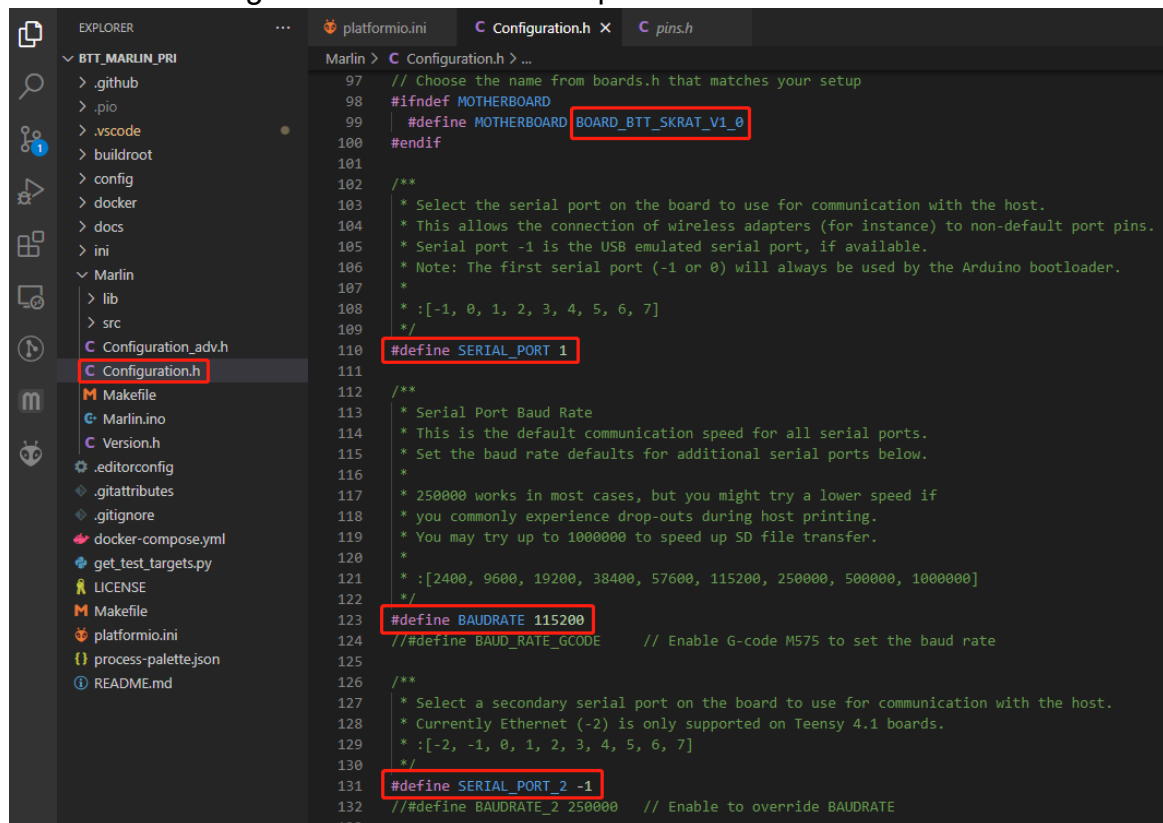
```
#define MOTHERBOARD BOARD_BTT_SKRAT_V1_0
```

```
#define SERIAL_PORT 1 (enable TFT serial port)
```

```
#define BAUDRATE 115200 (set baudrate to the same as the communication device)
```

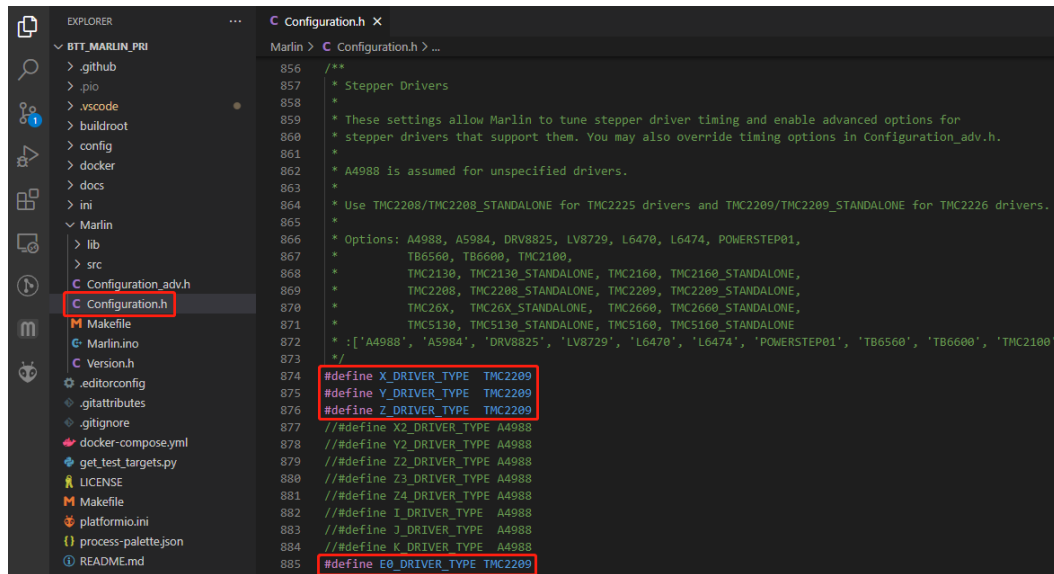
```
#define SERIAL_PORT_2 -1 (enable USB serial port)
```

The above settings can be enabled as required.



```
97 // Choose the name from boards.h that matches your setup
98 #ifndef MOTHERBOARD
99 #define MOTHERBOARD BOARD_BTT_SKRAT_V1_0
100 #endif
101
102 /**
103  * Select the serial port on the board to use for communication with the host.
104  * This allows the connection of wireless adapters (for instance) to non-default port pins.
105  * Serial port -1 is the USB emulated serial port, if available.
106  * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
107  *
108  * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
109  */
110 #define SERIAL_PORT 1
111
112 /**
113  * Serial Port Baud Rate
114  * This is the default communication speed for all serial ports.
115  * Set the baud rate defaults for additional serial ports below.
116  *
117  * 250000 works in most cases, but you might try a lower speed if
118  * you commonly experience drop-outs during host printing.
119  * You may try up to 1000000 to speed up SD file transfer.
120  *
121  * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
122  */
123 #define BAUDRATE 115200
124 // #define BAUD_RATE_GCODE // Enable G-code M575 to set the baud rate
125
126 /**
127  * Select a secondary serial port on the board to use for communication with the host.
128  * Currently Ethernet (-2) is only supported on Teensy 4.1 boards.
129  * :[-2, -1, 0, 1, 2, 3, 4, 5, 6, 7]
130  */
131 #define SERIAL_PORT_2 -1
132 // #define BAUDRATE_2 250000 // Enable to override BAUDRATE
133
```

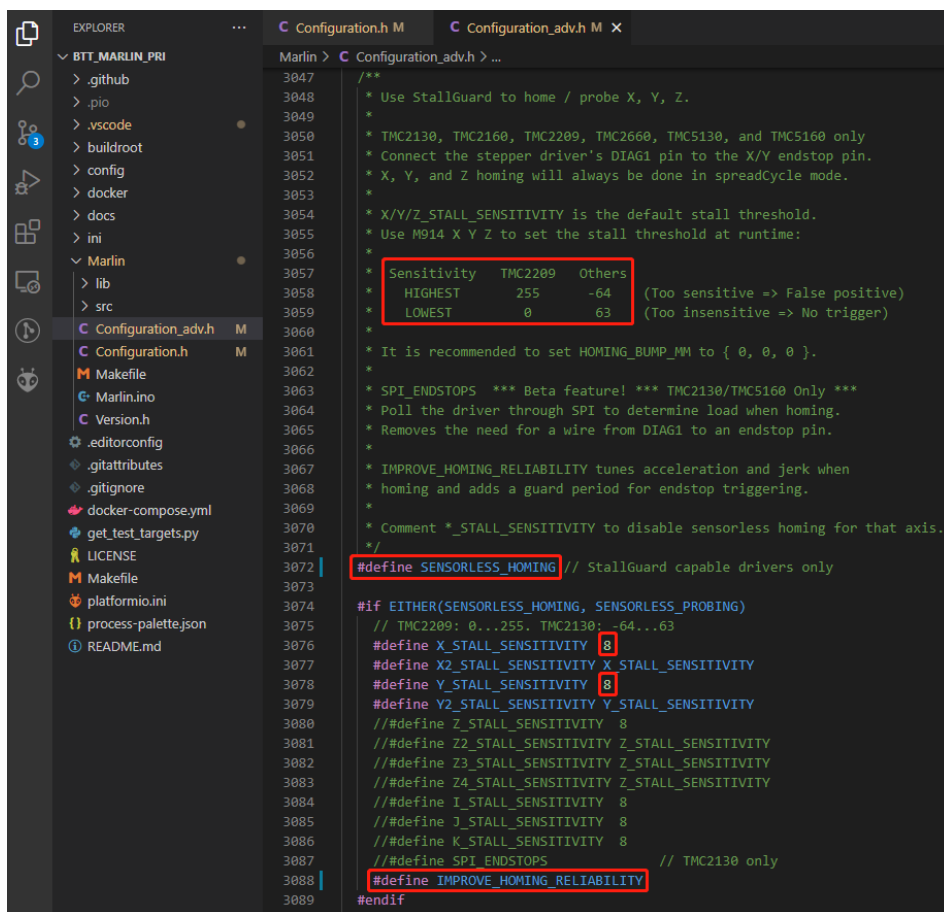

Configure Stepper Driver



```
856 /**
857  * Stepper Drivers
858  *
859  * These settings allow Marlin to tune stepper driver timing and enable advanced options for
860  * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
861  *
862  * A4988 is assumed for unspecified drivers.
863  *
864  * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STANDALONE for TMC2226 drivers.
865  *
866  * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
867  *          TB6560, TB6600, TMC2100,
868  *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
869  *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
870  *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
871  *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
872  * :["A4988", "A5984", "DRV8825", "LV8729", "L6470", "L6474", "POWERSTEP01", "TB6560", "TB6600", "TMC2100",
873  */
874 #define X_DRIVER_TYPE  TMC2209
875 #define Y_DRIVER_TYPE  TMC2209
876 #define Z_DRIVER_TYPE  TMC2209
877 // #define X2_DRIVER_TYPE  A4988
878 // #define Y2_DRIVER_TYPE  A4988
879 // #define Z2_DRIVER_TYPE  A4988
880 // #define Z3_DRIVER_TYPE  A4988
881 // #define Z4_DRIVER_TYPE  A4988
882 // #define I_DRIVER_TYPE  A4988
883 // #define J_DRIVER_TYPE  A4988
884 // #define K_DRIVER_TYPE  A4988
885 #define E0_DRIVER_TYPE TMC2209
```

Insert the jumper corresponding to the driver's mode and set the driver type in the firmware.

Sensorless Homing



```
3047 /**
3048  * Use StallGuard to home / probe X, Y, Z.
3049  *
3050  * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
3051  * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
3052  * X, Y, and Z homing will always be done in spreadCycle mode.
3053  *
3054  * X/Y/Z_STALL_SENSITIVITY is the default stall threshold.
3055  * Use M914 X Y Z to set the stall threshold at runtime:
3056  *
3057  * Sensitivity TMC2209 Others
3058  * HIGHEST 255 -64 (Too sensitive => False positive)
3059  * LOWEST 0 63 (Too insensitive => No trigger)
3060  *
3061  * It is recommended to set HOMING_BUMP_MM to { 0, 0, 0 }.
3062  *
3063  * SPI_ENDSTOPS *** Beta feature! *** TMC2130/TMC5160 Only ***
3064  * Poll the driver through SPI to determine load when homing.
3065  * Removes the need for a wire from DIAG1 to an endstop pin.
3066  *
3067  * IMPROVE_HOMING_RELIABILITY tunes acceleration and jerk when
3068  * homing and adds a guard period for endstop triggering.
3069  *
3070  * Comment *_STALL_SENSITIVITY to disable sensorless homing for that axis.
3071  */
3072 #define SENSORLESS_HOMING // StallGuard capable drivers only
3073
3074 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
3075 // TMC2209: 0...255. TMC2130: -64...63
3076 #define X_STALL_SENSITIVITY 8
3077 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
3078 #define Y_STALL_SENSITIVITY 8
3079 #define Y2_STALL_SENSITIVITY Y_STALL_SENSITIVITY
3080 // #define Z_STALL_SENSITIVITY 8
3081 // #define Z2_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3082 // #define Z3_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3083 // #define Z4_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3084 // #define I_STALL_SENSITIVITY 8
3085 // #define J_STALL_SENSITIVITY 8
3086 // #define K_STALL_SENSITIVITY 8
3087 // #define SPI_ENDSTOPS // TMC2130 only
3088 #define IMPROVE_HOMING_RELIABILITY
3089 #endif
```

```
#define SENSORLESS_HOMING // enable sensorless homing
```

```
#define xx_STALL_SENSITIVITY 8 // sensitivity setting, TMC2209 range from 0 to 255, higher number results in more sensitive trigger threshold, sensitivity too high will cause endpoint to trigger before gantry actually move to the end, lower number results in less sensitive trigger threshold, too low of sensitivity will cause endpoint to not trigger and gantrying continue.
```

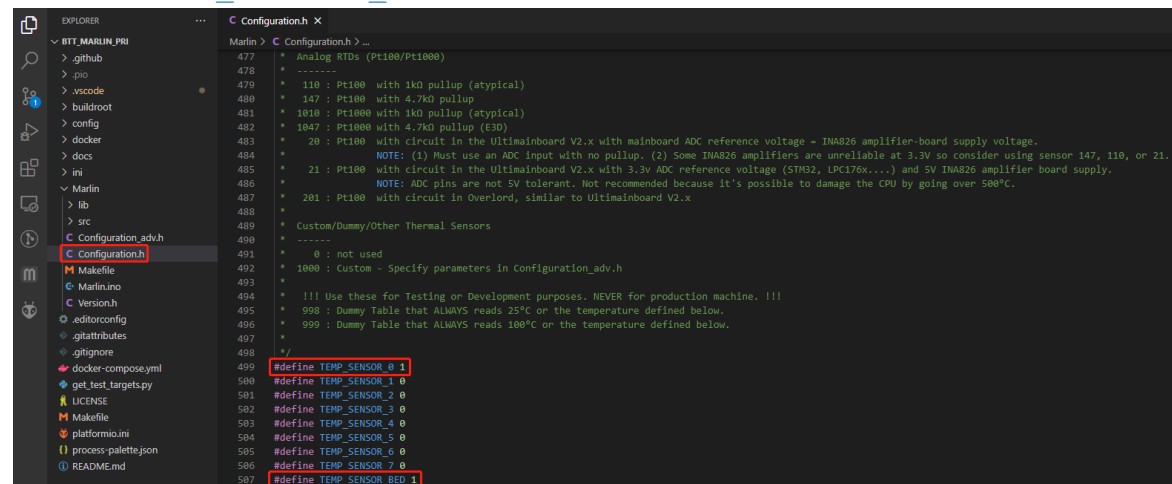
```
#define IMPROVE_HOMING_RELIABILITY // can be used to set independent motor current for homing moves(xx_CURRENT_HOME) to improve homing reliability.
```

100K NTC

In Marlin, 1 stands for 100K NTC +4.7K pull-up resistance.

```
#define TEMP_SENSOR_0 1
```

```
#define TEMP_SENSOR_BED 1
```



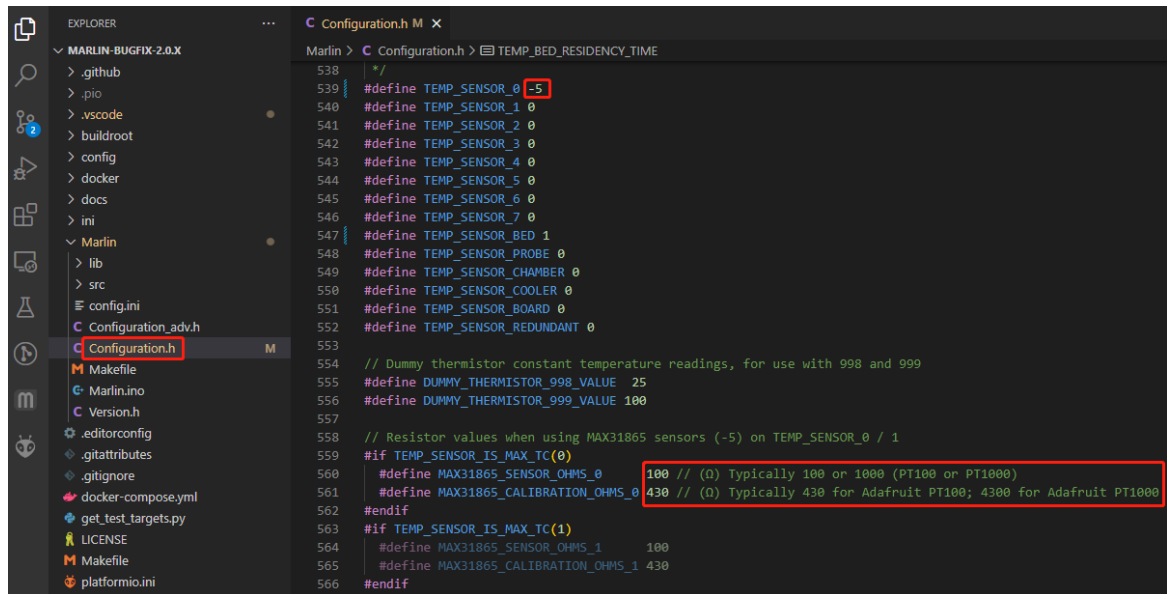
```
477 * Analog RTDs (Pt100/Pt1000)
478 * -----
479 * 110 : Pt100 with 1k0 pullup (atypical)
480 * 147 : Pt100 with 4.7k0 pullup
481 * 1010 : Pt1000 with 1k0 pullup (atypical)
482 * 1047 : Pt1000 with 4.7k0 pullup (E3D)
483 * 20 : Pt100 with circuit in the Ultimainboard V2.x with mainboard ADC reference voltage = INA826 amplifier-board supply voltage.
484 * NOTE: (1) Must use an ADC input with no pullup. (2) Some INA826 amplifiers are unreliable at 3.3V so consider using sensor 147, 110, or 21.
485 * 21 : Pt100 with circuit in the Ultimainboard V2.x with 3.3V ADC reference voltage (STM32, LPC176x....) and 5V INA826 amplifier board supply.
486 * NOTE: ADC pins are not 5V tolerant. Not recommended because it's possible to damage the CPU by going over 500°C.
487 * 281 : Pt100 with circuit in Overlord, similar to Ultimainboard V2.x
488 *
489 * Custom/Dummy/Other Thermal Sensors
490 * -----
491 * 0 : not used
492 * 1000 : Custom - Specify parameters in Configuration_adv.h
493 *
494 * !!! Use these for Testing or Development purposes. NEVER for production machine. !!!
495 * 998 : Dummy Table that ALWAYS reads 25°C on the temperature defined below.
496 * 999 : Dummy Table that ALWAYS reads 100°C on the temperature defined below.
497 *
498 */
499 #define TEMP_SENSOR_0 1
500 #define TEMP_SENSOR_1 0
501 #define TEMP_SENSOR_2 0
502 #define TEMP_SENSOR_3 0
503 #define TEMP_SENSOR_4 0
504 #define TEMP_SENSOR_5 0
505 #define TEMP_SENSOR_6 0
506 #define TEMP_SENSOR_7 0
507 #define TEMP_SENSOR_BED 1
```

Max31865

In Marlin, -5 stands for MAX31865

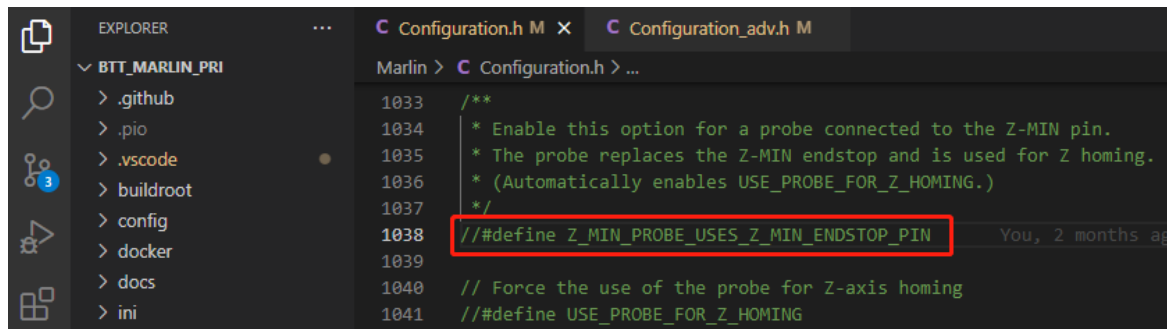
```
#define MAX31865_SENSOR_OHMS_0 100 // PT100:100, PT1000:1000
```

```
#define MAX31865_CALIBRATION_OHMS_0 430 // PT100:430, PT1000:4300
```



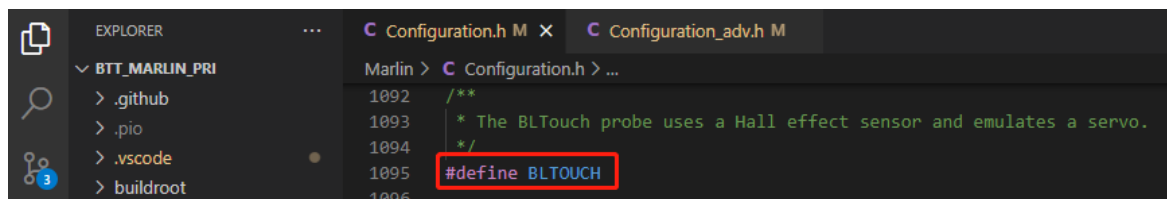
A screenshot of the Visual Studio Code editor showing the 'Configuration.h' file in the 'MARLIN-BUGFIX-2.0.X' project. The file contains various preprocessor definitions for temperature sensors. Two lines are highlighted with red boxes: line 539, '#define TEMP_SENSOR_0 -5', and line 561, '#define MAX31865_CALIBRATION_OHMS_0 430 // (Ω) Typically 430 for Adafruit PT100; 4300 for Adafruit PT1000'. The Explorer sidebar on the left shows the project structure, with 'Configuration.h' selected.

BLTouch



A screenshot of the Visual Studio Code editor showing the 'Configuration.h' file in the 'BTT_MARLIN_PRI' project. The file contains definitions for BLTouch. Two lines are highlighted with red boxes: line 1038, '//#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN', and line 1095, '#define BLTOUCH'. The Explorer sidebar on the left shows the project structure, with 'Configuration.h' selected.

```
//#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN //
```



A screenshot of the Visual Studio Code editor showing the 'Configuration.h' file in the 'BTT_MARLIN_PRI' project. The file contains definitions for BLTouch. One line is highlighted with a red box: line 1095, '#define BLTOUCH'. The Explorer sidebar on the left shows the project structure, with 'Configuration.h' selected.

```
#define BLTOUCH // Enable BLTouch
```

```

1182 * Some examples:
1183 * #define NOZZLE_TO_PROBE_OFFSET { 10, 10, -1 } // Example "1"
1184 * #define NOZZLE_TO_PROBE_OFFSET {-10, 5, -1 } // Example "2"
1185 * #define NOZZLE_TO_PROBE_OFFSET { 5, -5, -1 } // Example "3"
1186 * #define NOZZLE_TO_PROBE_OFFSET {-15,-10, -1 } // Example "4"
1187 *
1188 * +-+ BACK +-+
1189 * | | [+ ] |
1190 * L | 1 | R <-- Example "1" (right+, back+)
1191 * E | 2 | I <-- Example "2" ( left-, back+)
1192 * F |[-] N [+ ] G <-- Nozzle
1193 * T | 3 | H <-- Example "3" (right+, front-)
1194 * | 4 | T <-- Example "4" ( left-, front-)
1195 * | | [- ] |
1196 * 0-- FRONT --+
1197 */
1198 #define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 }
1199
1200 // Most probes should stay away from the edges of the bed, but
1201 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
1202 #define PROBING_MARGIN 10
1203
1204 // X and Y axis travel speed (mm/min) between probes
1205 #define XY_PROBE_FEEDRATE (133*60)
1206
1207 // Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_PROBING == 2)
1208 #define Z_PROBE_FEEDRATE_FAST (4*60)
1209
1210 // Feedrate (mm/min) for the "accurate" probe of each point
1211 #define Z_PROBE_FEEDRATE_SLOW (Z_PROBE_FEEDRATE_FAST / 2)
1212

```

`#define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 }` // set BLTouch probe offset

`#define PROBING_MARGIN 10` // set distance between probe area and print area perimeter

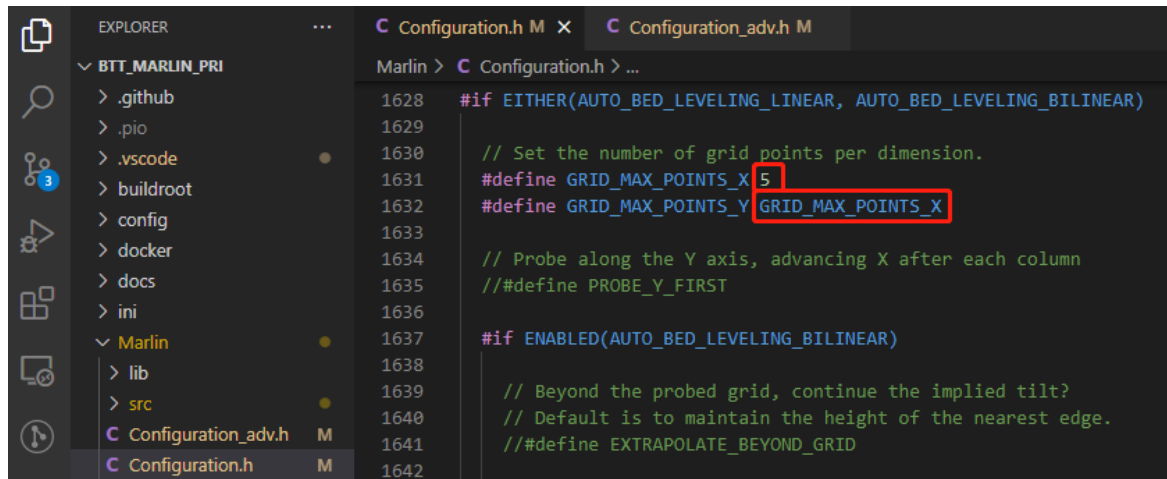
```

1562 //#define AUTO_BED_LEVELING_3POINT
1563 //#define AUTO_BED_LEVELING_LINEAR
1564 #define AUTO_BED_LEVELING_BILINEAR
1565 //#define AUTO_BED_LEVELING_UBL
1566 //#define MESH_BED_LEVELING
1567
1568 /**
1569  * Normally G28 leaves leveling disabled on completion. Enable one of
1570  * these options to restore the prior leveling state or to always enable
1571  * leveling immediately after G28.
1572  */
1573 //#define RESTORE_LEVELING_AFTER_G28
1574 #define ENABLE_LEVELING_AFTER_G28
1575
1576 /**

```

`#define AUTO_BED_LEVELING_BILINEAR` // set probe pattern

`#define RESTORE_LEVELING_AFTER_G28` // apply leveling after G28 homing command

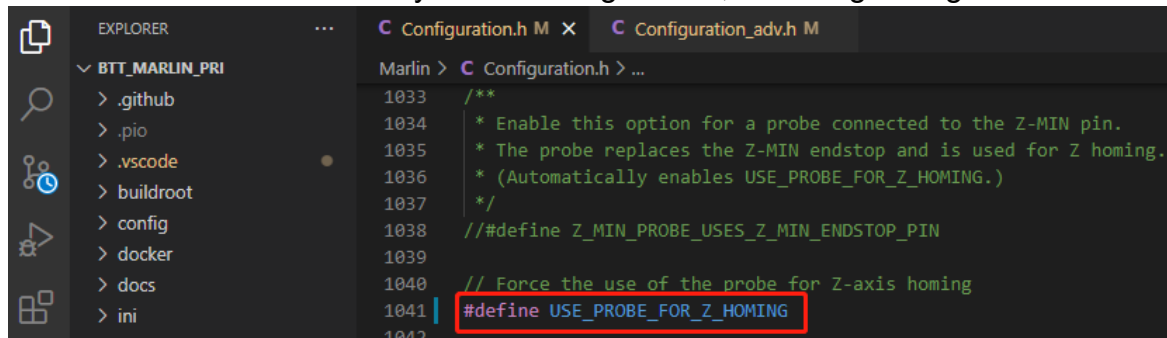


```
1628 #if EITHER(AUTO_BED_LEVELING_LINEAR, AUTO_BED_LEVELING_BILINEAR)
1629
1630 // Set the number of grid points per dimension.
1631 #define GRID_MAX_POINTS_X 5
1632 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
1633
1634 // Probe along the Y axis, advancing X after each column
1635 // #define PROBE_Y_FIRST
1636
1637 #if ENABLED(AUTO_BED_LEVELING_BILINEAR)
1638
1639 // Beyond the probed grid, continue the implied tilt?
1640 // Default is to maintain the height of the nearest edge.
1641 // #define EXTRAPOLATE_BEYOND_GRID
1642
```

`#define GRID_MAX_POINTS_X 5` // set number of probe points for X axis, usually 5 point is sufficient

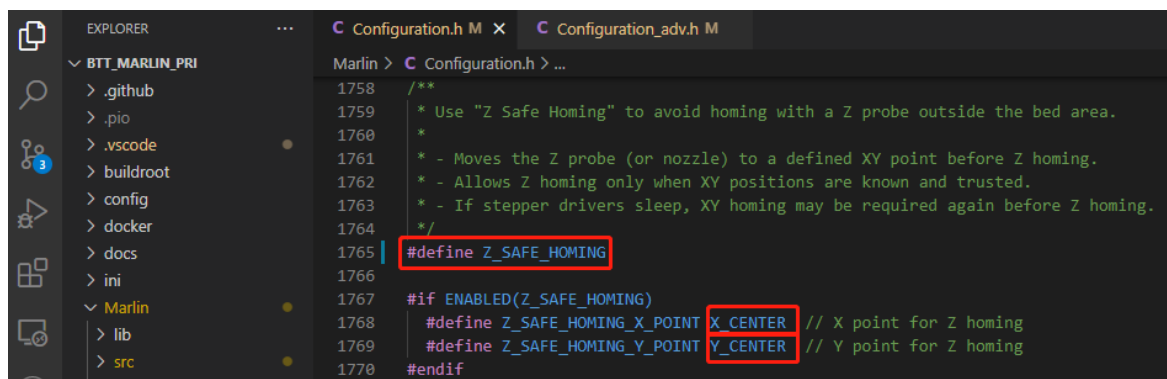
`#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X` // set number of probe points for Y axis to the same as X axis

If BLTouch also functions as your Z homing sensor, no wiring change is needed.



```
1033 /**
1034  * Enable this option for a probe connected to the Z-MIN pin.
1035  * The probe replaces the Z-MIN endstop and is used for Z homing.
1036  * (Automatically enables USE_PROBE_FOR_Z_HOMING.)
1037  */
1038 // #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
1039
1040 // Force the use of the probe for Z-axis homing
1041 #define USE_PROBE_FOR_Z_HOMING
1042
```

`#define USE_PROBE_FOR_Z_HOMING` // use Z Probe(BLTouch) for Z homing.



```
1758 /**
1759  * Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
1760  *
1761  * - Moves the Z probe (or nozzle) to a defined XY point before Z homing.
1762  * - Allows Z homing only when XY positions are known and trusted.
1763  * - If stepper drivers sleep, XY homing may be required again before Z homing.
1764  */
1765 #define Z_SAFE_HOMING
1766
1767 #if ENABLED(Z_SAFE_HOMING)
1768 #define Z_SAFE_HOMING_X_POINT X_CENTER // X point for Z homing
1769 #define Z_SAFE_HOMING_Y_POINT Y_CENTER // Y point for Z homing
1770 #endif
```

`#define Z_SAFE_HOMING` // home Z at the center of print bed to prevent probing outside of the print bed

Auto Power Off (Relay V1.2)

```

359 /**
360  * Power Supply Control
361  *
362  * Enable and connect the power supply to the PS_ON_PIN.
363  * Specify whether the power supply is active HIGH or active LOW.
364  */
365 #define PSU_CONTROL
366 #define PSU_NAME "Power Supply"
367
368 #if ENABLED(PSU_CONTROL)
369 // #define MKS_PWC // Using the MKS PWC add-on
370 // #define PS_OFF_CONFIRM // Confirm dialog when power off
371 // #define PS_OFF_SOUND // Beep 1s when power off
372 #define PSU_ACTIVE_STATE HIGH // Set 'LOW' for ATX, 'HIGH' for X-Box
373
374 // #define PSU_DEFAULT_OFF // Keep power off until enabled directly with M80
375 // #define PSU_POWERUP_DELAY 250 // (ms) Delay for the PSU to warm up to full power
376
377 // #define POWER_OFF_TIMER // Enable M81 D<seconds> to power off after a delay
378 // #define POWER_OFF_WAIT_FOR_COOLDOWN // Enable M81 S to power off only after cooldown

```

`#define PSU_CONTROL` // enable PSU control to turn on and off using M80 and M81

`#define PSU_ACTIVE_STATE HIGH` // set turn on level, Relay V1.2 is turned on with high level and turned off with low level, so this setting needs to be HIGH.

Power Loss Recovery

There are two methods for power loss recovery:

1. No extra module is needed, the motherboard will write the current print status to the microSD card after every layer is printed, which shortens the life of the microSD card severely.

```

1459 * Store the current state to the SD card at the start of each layer
1460 * during SD printing. If the recovery file is found at boot time, present
1461 * an option on the LCD screen to continue the print from the last-known
1462 * point in the file.
1463 */
1464 #define POWER_LOSS_RECOVERY
1465 #if ENABLED(POWER_LOSS_RECOVERY)
1466 #define PLR_ENABLED_DEFAULT true // Power Loss Recovery enabled by default. (Set with 'M13 S0' & M500)
1467 // #define BACKUP_POWER_SUPPLY // Backup power / UPS to move the steppers on power loss
1468 #define POWER_LOSS_ZRAISE 10 // (mm) Z axis raise on resume (on power loss with UPS)
1469 // #define POWER_LOSS_PIN 44 // Pin to detect power loss. Set to -1 to disable default pin on boards without module.
1470 // #define POWER_LOSS_STATE HIGH // State of pin indicating power loss
1471 // #define POWER_LOSS_PULLUP // Set pullup / pulldown as appropriate for your sensor
1472 // #define POWER_LOSS_PULLDOWN
1473 #define POWER_LOSS_PURGE_LEN 20 // (mm) Length of filament to purge on resume
1474 #define POWER_LOSS_RETRACT_LEN 10 // (mm) Length of filament to retract on fail. Requires backup power.
1475
1476 // Without a POWER_LOSS_PIN the following option helps reduce wear on the SD card,
1477 // especially with "vase mode" printing. Set too high and vases cannot be continued.
1478 #define POWER_LOSS_MIN_Z_CHANGE 0.05 // (mm) Minimum Z change before saving power-loss data
1479
1480 // Enable if Z homing is needed for proper recovery. 99.9% of the time this should be disabled!
1481 // #define POWER_LOSS_RECOVER_ZHOME
1482 #if ENABLED(POWER_LOSS_RECOVER_ZHOME)
1483 // #define POWER_LOSS_ZHOME_POS { 0, 0 } // Safe XY position to home Z while avoiding objects on the bed
1484 #endif
1485 #endif

```

`#define POWER_LOSS_RECOVERY` // enable power loss recovery
`#define PLR_ENABLED_DEFAULT true` // true default to power loss recovery enabled

2. External UPS 24V V1.0 module, when power is cut, the module will provide power to the board and signal the board to save the current print status to the microSD card. This method has virtually no effect on the life of the microSD

card.

```

1459 * Store the current state to the SD Card at the start of each layer
1460 * during SD printing. If the recovery file is found at boot time, present
1461 * an option on the LCD screen to continue the print from the last-known
1462 * point in the file.
1463 */
1464 #define POWER_LOSS_RECOVERY
1465 #if ENABLED(POWER_LOSS_RECOVERY)
1466 #define PLR_ENABLED_DEFAULT true // Power Loss Recovery enabled by default. (Set with 'M13 Sn' & M50)
1467 #define BACKUP_POWER_SUPPLY // Backup power / UPS to move the steppers on power loss
1468 #define POWER_LOSS_ZRAISE 10 // (mm) Z axis raise on resume (on power loss with UPS)
1469 // #define POWER_LOSS_PIN -1 // Pin to detect power loss. Set to -1 to disable default pin on boards without module.
1470 #define POWER_LOSS_STATE HIGH // State of pin indicating power loss
1471 #define POWER_LOSS_PULLUP // Set pullup / pulldown as appropriate for your sensor
1472 // #define POWER_LOSS_PULLDOWN
1473 #define POWER_LOSS_PURGE_LEN 20 // (mm) Length of filament to purge on resume
1474 #define POWER_LOSS_RETRACT_LEN 10 // (mm) Length of filament to retract on fail. Requires backup power.
1475
1476 // Without a POWER_LOSS_PIN the following option helps reduce wear on the SD card,
1477 // especially with "vase mode" printing. Set too high and vases cannot be continued.
1478 #define POWER_LOSS_MIN_Z_CHANGE 0.05 // (mm) Minimum Z change before saving power-loss data
1479
1480 // Enable if Z homing is needed for proper recovery. 99.9% of the time this should be disabled!
1481 // #define POWER_LOSS_RECOVER_ZHOME
1482 #if ENABLED(POWER_LOSS_RECOVER_ZHOME)
1483 // #define POWER_LOSS_ZHOME_POS { 0, 0 } // Safe XY position to home Z while avoiding objects on the bed
1484 #endif
1485 #endif

```

#define POWER_LOSS_RECOVERY // enable power loss recovery
#define PLR_ENABLED_DEFAULT true // true default to power loss recovery enabled
#define POWER_LOSS_ZRAISE 10 // raise the print head by 10mm after power loss to prevent the nozzle from touching the printed part
#define POWER_LOSS_STATE HIGH // set signal level, UPS 24V V1.0 returns a low level when not triggered and a HIGH level when power is cut, thus this setting needs to be HIGH.

RGB

```

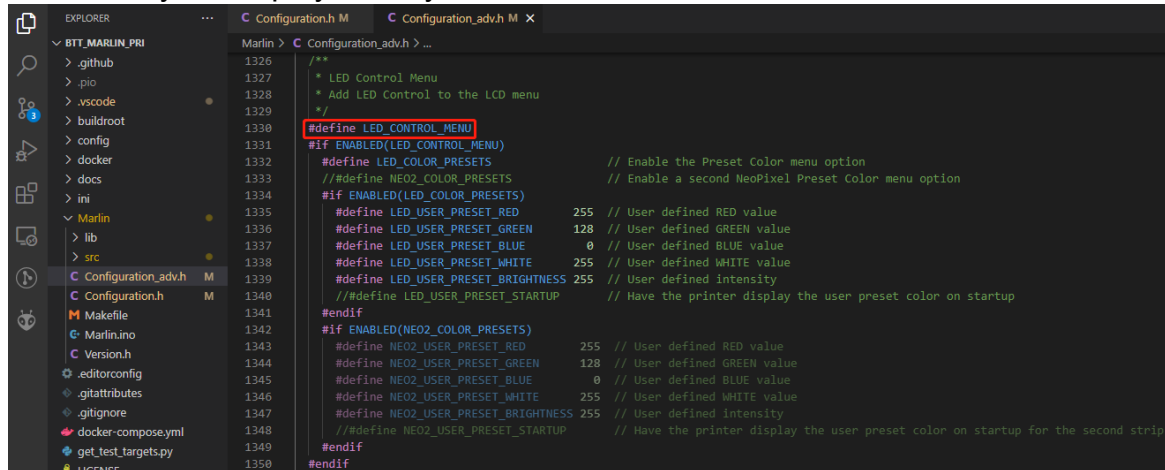
2926 // Support for Adafruit NeoPixel LED driver
2927 #define NEOPIXEL_LED
2928 #if ENABLED(NEOPIXEL_LED)
2929 #define NEOPIXEL_TYPE NEO_GRB // NEO_GRBW / NEO_GRB - four/three channel driver type (defined in Adafruit_NeoPixel.h)
2930 // #define NEOPIXEL_PIN 4 // LED driving pin
2931 // #define NEOPIXEL2_TYPE NEOPIXEL_TYPE
2932 // #define NEOPIXEL2_PIN 5
2933 #define NEOPIXEL_PIXELS 30 // Number of LEDs in the strip. (Longest strip when NEOPIXEL2_SEPARATE is disabled.)
2934 #define NEOPIXEL_IS_SEQUENTIAL // Sequential display for temperature change - LED by LED. Disable to change all LEDs at once.
2935 #define NEOPIXEL_BRIGHTNESS 255 // Initial brightness (0-255)
2936 #define NEOPIXEL_STARTUP_TEST // Cycle through colors at startup
2937
2938 // Support for second Adafruit NeoPixel LED driver controlled with M150 S1 ...
2939 // #define NEOPIXEL2_SEPARATE
2940 #if ENABLED(NEOPIXEL2_SEPARATE)
2941 #define NEOPIXEL2_PIXELS 15 // Number of LEDs in the second strip
2942 #define NEOPIXEL2_BRIGHTNESS 127 // Initial brightness (0-255)
2943 #define NEOPIXEL2_STARTUP_TEST // Cycle through colors at startup
2944 #else
2945 // #define NEOPIXEL2_INSERTS // Default behavior is NeoPixel 2 in parallel
2946 #endif
2947
2948 // Use some of the NeoPixel LEDs for static (background) lighting
2949 // #define NEOPIXEL_BKGD_INDEX_FIRST 0 // Index of the first background LED
2950 // #define NEOPIXEL_BKGD_INDEX_LAST 5 // Index of the last background LED
2951 // #define NEOPIXEL_BKGD_COLOR { 255, 255, 255, 0 } // R, G, B, W
2952 // #define NEOPIXEL_BKGD_ALWAYS_ON // Keep the backlight on when other NeoPixels are off
2953 #endif

```

#define NEOPIXEL_LED // enable Neopixel
#define NEOPIXEL_TYPE NEO_GRB // set Neopixel type
// #define NEOPIXEL_PIN 4 // disable PIN setting, use the correct signal pin in the pin file of the motherboard
#define NEOPIXEL_PIXELS 30 // number of LEDs

`#define NEOPIXEL_STARTUP_TEST` // the light will show red green and blue sequentially to self-test.

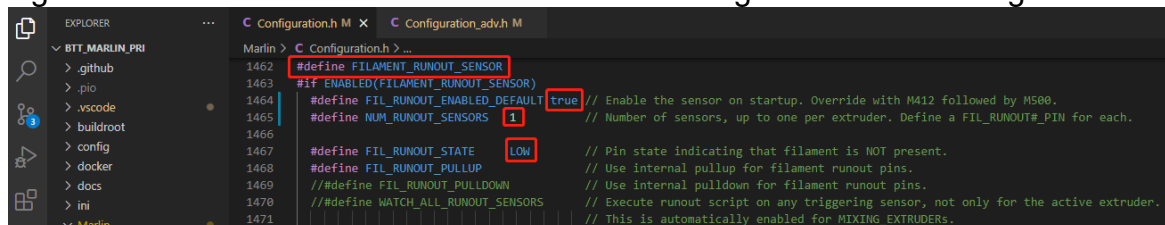
If you are using displays like CR10_STOCKDISPLAY, etc., you can also control RGB from your display directly.



`#define LED_CONTROL_MENU` // Add LED control to your menu

Filament Sensor

Standard filament run out sensors are usually comprised of a micro switch that signals the motherboard of filament status with a High or a Low level signal.



`#define FILAMENT_RUNOUT_SENSOR` // enable filament run out sensor
`#define FIL_RUNOUT_ENABLED_DEFAULT true` // true default to filament run out sensor enabled
`#define NUM_RUNOUT_SENSORS 1` // number of filament run out sensor
`#define FIL_RUNOUT_STATE LOW` // voltage level of the filament run out sensor trigger signal.

Smart Filament Sensor(SFS V1.0)

The smart filament sensor works by continuously sending signals to the motherboard to communicate filament status.

Shenzhen Big Tree Technology Co., Ltd.
BIGTREETECH

```
1462 #define FILAMENT_RUNOUT_SENSOR
1463 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1464 #define FIL_RUNOUT_ENABLED_DEFAULT true // Enable the sensor on startup. Override with M412 followed by M500.
1465 #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to one per extruder. Define a FIL_RUNOUT#_PIN for each.
1466
1467 #define FIL_RUNOUT_STATE LOW // Pin state indicating that filament is NOT present.
1468 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins...
1472
1473 // Override individually if the runout sensors vary...
1477
1478 // #define FIL_RUNOUT2_STATE LOW...
1481
1482 // #define FIL_RUNOUT3_STATE LOW...
1485
1486 // #define FIL_RUNOUT4_STATE LOW...
1489
1490 // #define FIL_RUNOUT5_STATE LOW...
1493
1494 // #define FIL_RUNOUT6_STATE LOW...
1497
1498 // #define FIL_RUNOUT7_STATE LOW...
1501
1502 // #define FIL_RUNOUT8_STATE LOW...
1505
1506 // Commands to execute on filament runout.
1507 // With multiple runout sensors use the %c placeholder for the current tool in commands (e.g., "M600 T%c")
1508 // NOTE: After 'M412 H1' the host handles filament runout and this script does not apply.
1509 #define FILAMENT_RUNOUT_SCRIPT "M600"
1510
1511 // After a runout is detected, continue printing this length of filament
1512 // before executing the runout script. Useful for a sensor at the end of
1513 // a feed tube. Requires 4 bytes SRAM per sensor, plus 4 bytes overhead.
1514 #define FILAMENT_RUNOUT_DISTANCE_MM 7
1515
1516 #ifdef FILAMENT_RUNOUT_DISTANCE_MM
1517 // Enable this option to use an encoder disc that toggles the runout pin
1518 // as the filament moves. (Be sure to set FILAMENT_RUNOUT_DISTANCE_MM
1519 // large enough to avoid false positives.)
1520 #define FILAMENT_MOTION_SENSOR
1521 #endif
1522 #endif
```

`#define FILAMENT_MOTION_SENSOR` // set encoder type
`#define FILAMENT_RUNOUT_DISTANCE_MM 7` // set sensitivity, SFS V1.0
nominal setting should be 7mm, which means if no signal of filament movement is detected after 7mm of filament travel command, filament error will be triggered.

The settings below also need to be set to instruct the printer to park the nozzle after a filament error is detected.

```
1907 #define NOZZLE_PARK_FEATURE
1908
1909 #if ENABLED(NOZZLE_PARK_FEATURE)
1910 // Specify a park position as { X, Y, Z raise }
1911 #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
1912 // #define NOZZLE_PARK_X_ONLY // X move only is required to park
1913 // #define NOZZLE_PARK_Y_ONLY // Y move only is required to park
1914 #define NOZZLE_PARK_Z_RAISE_MIN 2 // (mm) Always raise Z by at least this distance
1915 #define NOZZLE_PARK_XY_FEEDRATE 100 // (mm/s) X and Y axes feedrate (also used for delta Z axis)
1916 #define NOZZLE_PARK_Z_FEEDRATE 5 // (mm/s) Z axis feedrate (not used for delta printers)
1917 #endif
```

`#define NOZZLE_PARK_FEATURE` // park nozzle
`#define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }`
// set the X,Y, and Z offset coordinate of the nozzle

```
2488 * Requirements:
2489 * - For Filament Change parking enable and configure NOZZLE_PARK_FEATURE.
2490 * - For user interaction enable an LCD display, HOST_PROMPT_SUPPORT, or EMERGENCY_PARSER.
2491 *
2492 * Enable PARK_HEAD_ON_PAUSE to add the G-code M125 Pause and Park.
2493 */
2494 #define ADVANCED_PAUSE_FEATURE
2495 #if ENABLED(ADVANCED_PAUSE_FEATURE)
```

`#define ADVANCED_PAUSE_FEATURE` // retraction setting of nozzle park movement and filament purge distance after the print is resumed

Compile Firmware

1. Click "✓" to compile the firmware.



2. Copy the compiled "firmware.bin" to microSD card and insert to motherboard to update firmware.

```
Checking size .pio\build\STM32F401RC_btt\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [===      ] 28.8% (used 18872 bytes from 65536 bytes)
Flash: [===== ] 74.6% (used 195436 bytes from 262144 bytes)
Building .pio\build\STM32F401RC_btt\firmware.bin
===== [SUCCESS] Took 90.28 seconds

Environment      Status      Duration
-----
STM32F401RC_btt  SUCCESS     00:01:30.275
===== 1 succeeded in 00:01:30.275

Terminal will be reused by tasks, press any key to close it.
```

Klipper

Using Raspberry Pi

Download OS Image

When using CM4, download the image of FluidD, Mainsail directly, also, you can download a pure OS image from the Raspberry Pi official website:

FluidD: <https://github.com/fluidD-core/FluidDPI/releases>

Mainsail: <https://github.com/mainsail-crew/MainsailOS/releases>

Official Raspberry Pi OS Image: <https://www.raspberrypi.com/software/operating-systems>

(The usage of CM4 is slightly different from the standard Raspberry Pi 3B, 4B, etc., CM4 needs to refer to the system settings section to enable the system's USB, DSI, and other interfaces).

Raspberry Pi OS

Our recommended operating system for most users.

Compatible with:

All Raspberry Pi models

Raspberry Pi OS with desktop

Release date: January 28th 2022
System: 32-bit
Kernel version: 5.10
Debian version: 11 (bullseye)
Size: 1,246MB
[Show SHA256 file integrity hash:](#)
[Release notes](#)

Download

[Download torrent](#)
[Archive](#)

Raspberry Pi OS with desktop and recommended software

Release date: January 28th 2022
System: 32-bit
Kernel version: 5.10
Debian version: 11 (bullseye)
Size: 3,267MB
[Show SHA256 file integrity hash:](#)
[Release notes](#)

Download

[Download torrent](#)
[Archive](#)

Raspberry Pi OS Lite

Release date: January 28th 2022
System: 32-bit
Kernel version: 5.10
Debian version: 11 (bullseye)
Size: 482MB
[Show SHA256 file integrity hash:](#)
[Release notes](#)

Download

[Download torrent](#)
[Archive](#)

Download and Install Raspberry Pi Imager

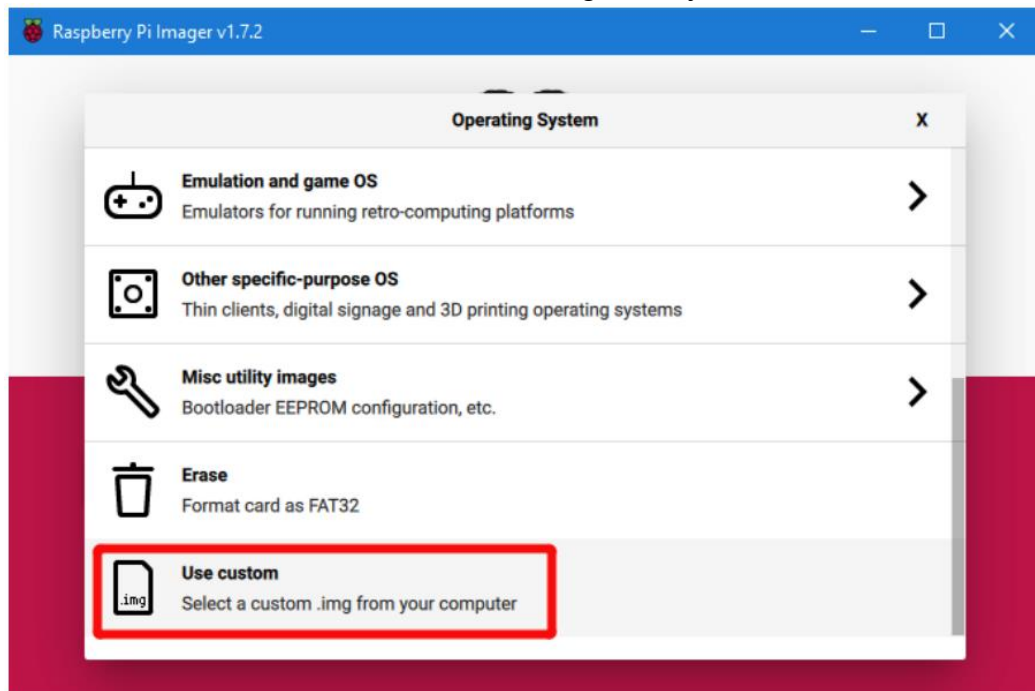
Install the official Raspberry Pi Imager <https://www.raspberrypi.com/software/>

Write Image

1. Insert a microSD card into your computer.
2. Choose OS.



3. Select "Use custom", then select the image that you downloaded.



4. Click the settings icon in the lower right corner.



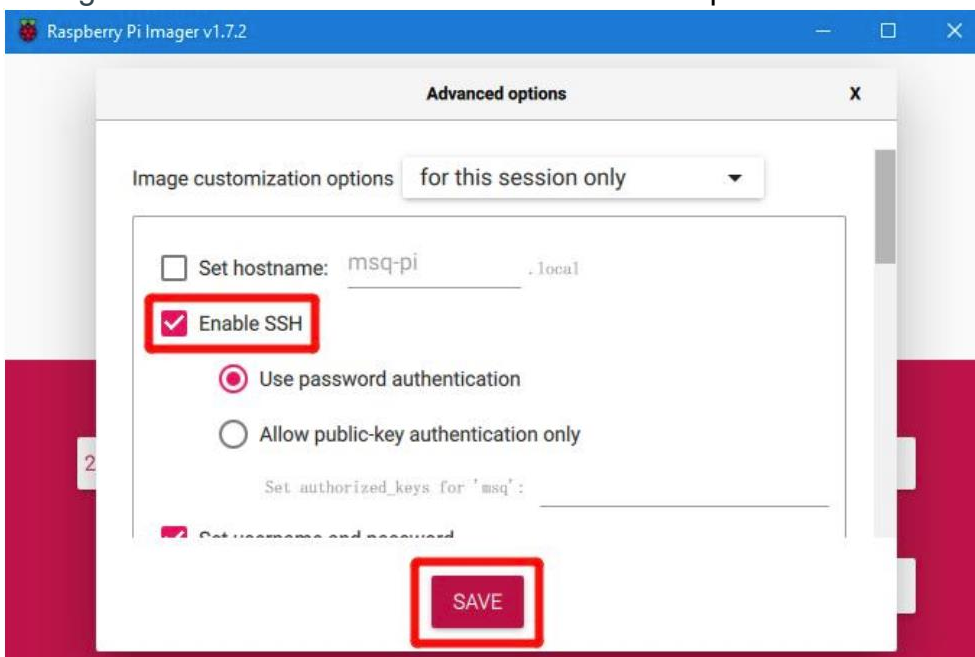
5. "Enable SSH" and then click "Save", there are other functions that can be set in this interface, please modify them according to your needs. Details are as follows:

Set hostname: raspberrypi.local // custom hostname, default is raspberrypi.local

Enable SSH

Set username and password // custom username and password, default username: pi, password: raspberry

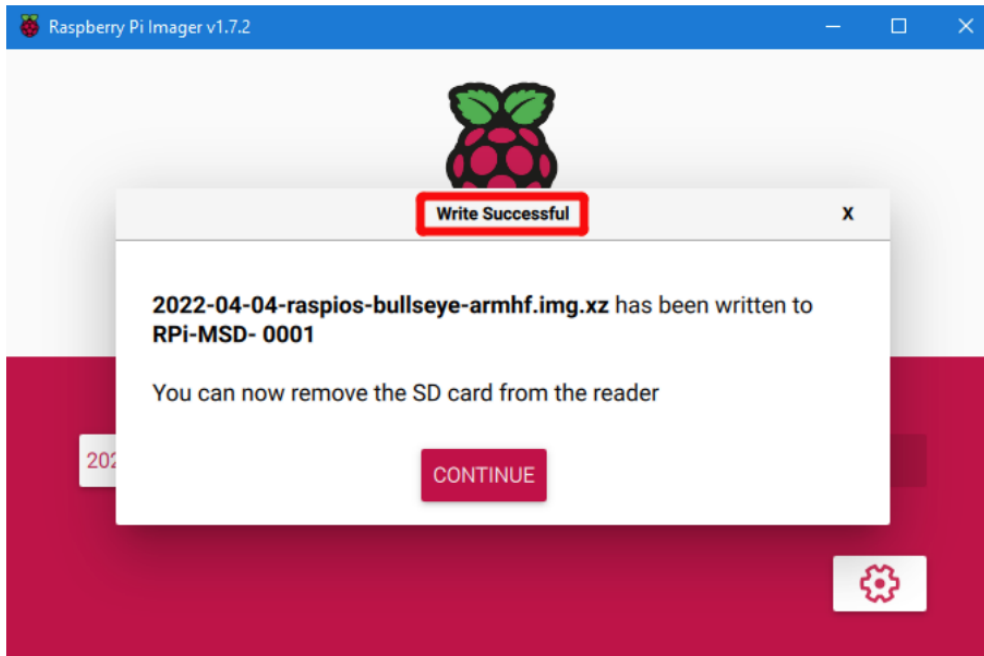
Configure wireless LAN // custom WiFi name and password



6. Select the microSD card and click "WRITE" (WRITE the image will format the microSD card. Be careful not to select the wrong storage device, otherwise the data will be formatted).



7. Wait for the writing to finish.



Using BIGTREETECH CB1

Download OS Image

When using CB1, please download and install the OS image provided by BIGTREETECH.

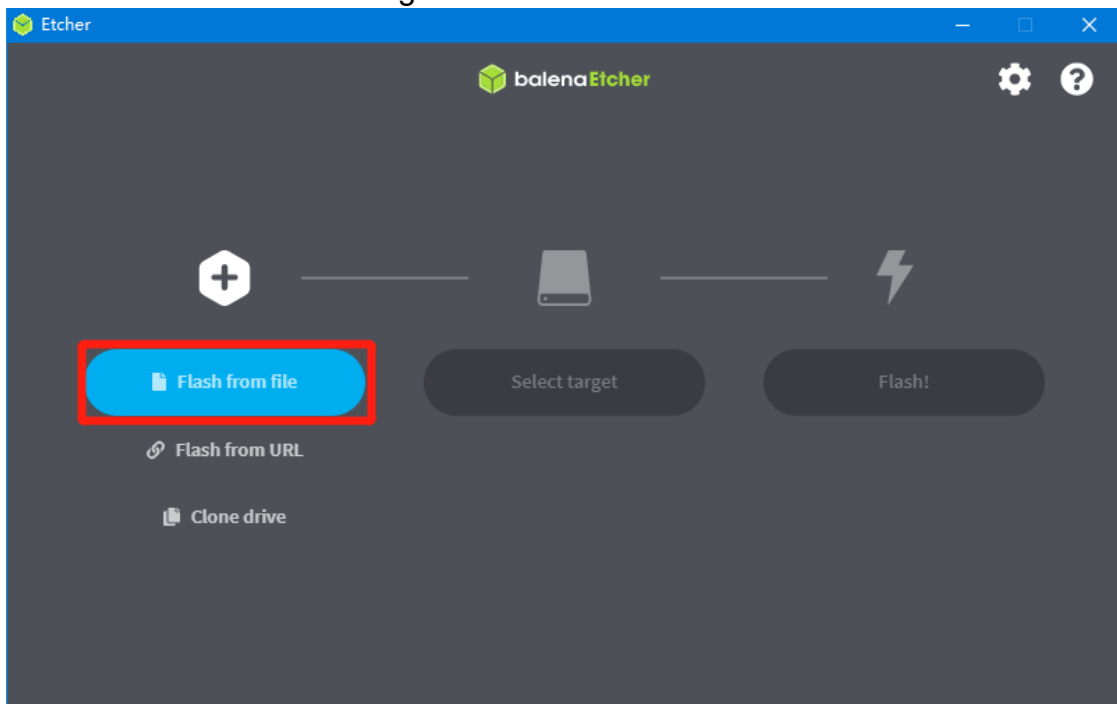
<https://github.com/bigtreotech/CB1/releases>

Download and Install balenaEtcher

balenaEtcher: <https://www.balena.io/etcher/>

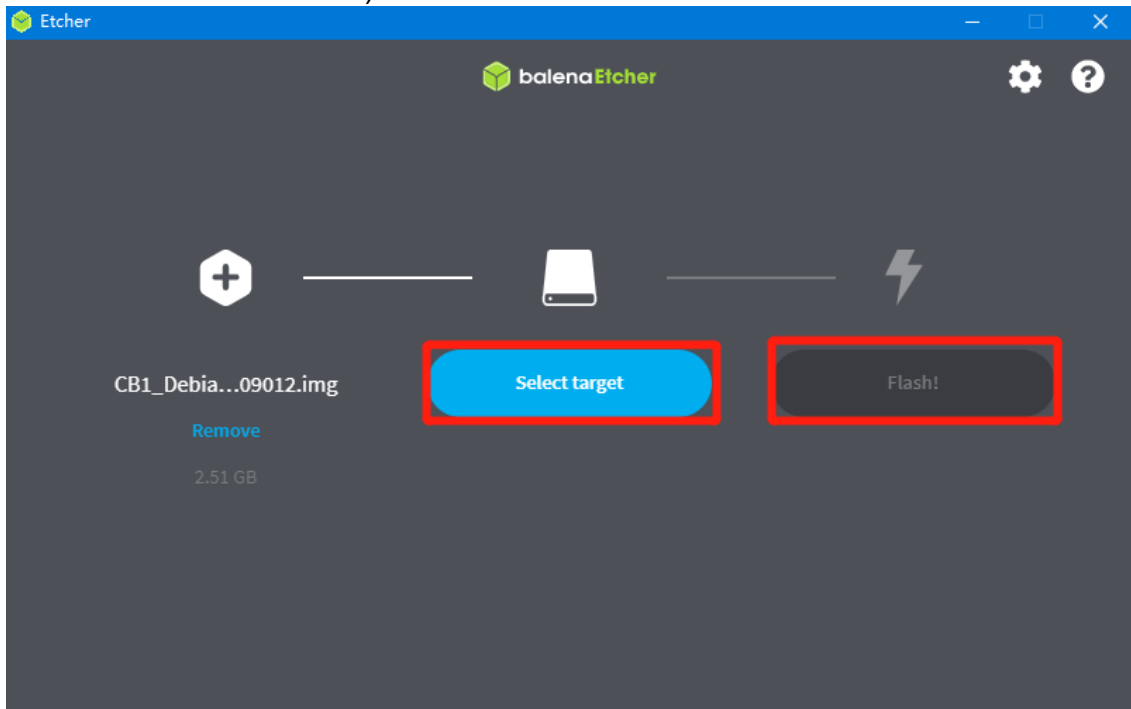
Write OS

1. Insert a microSD card into your computer via a card reader.
2. Select the downloaded image.

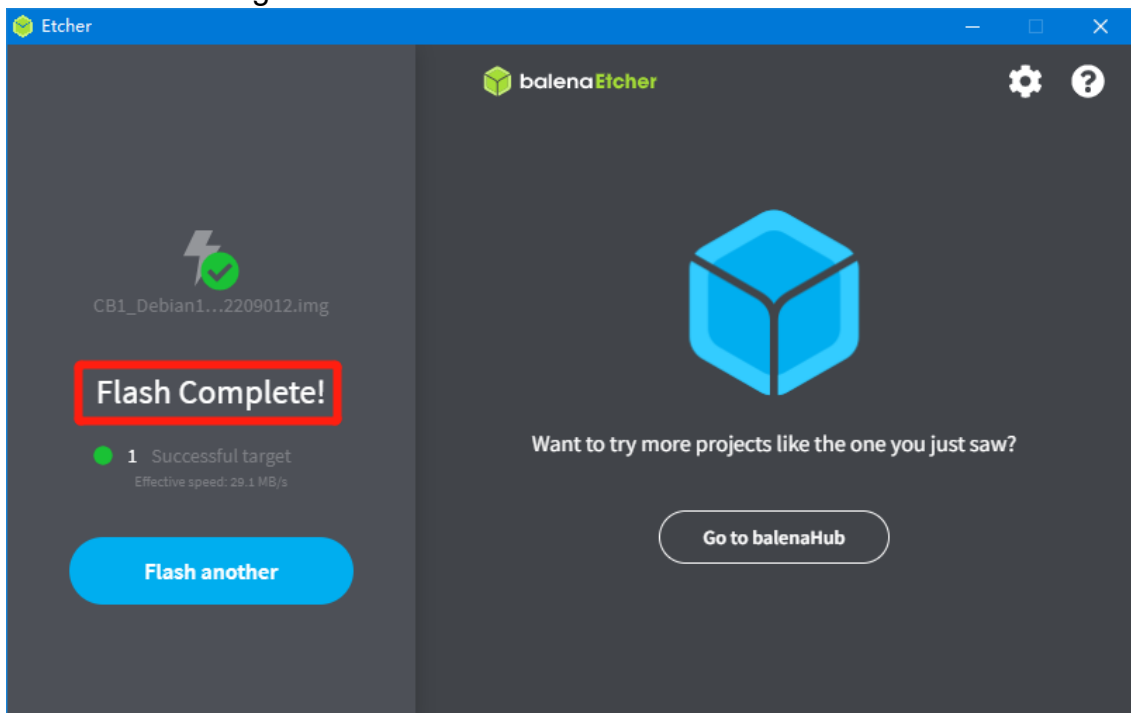


3. Select the microSD card and click "WRITE" (WRITE the image will format the MicroSD card. Be careful not to select the wrong storage device, otherwise

the data will be formatted).



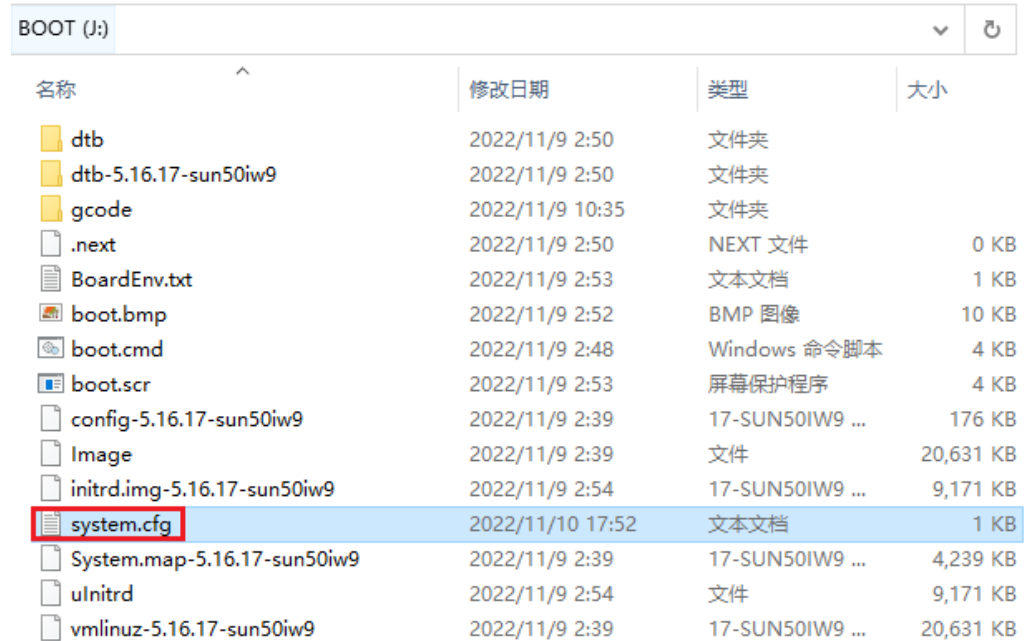
4. Wait for the writing to finish.



WiFi Setting

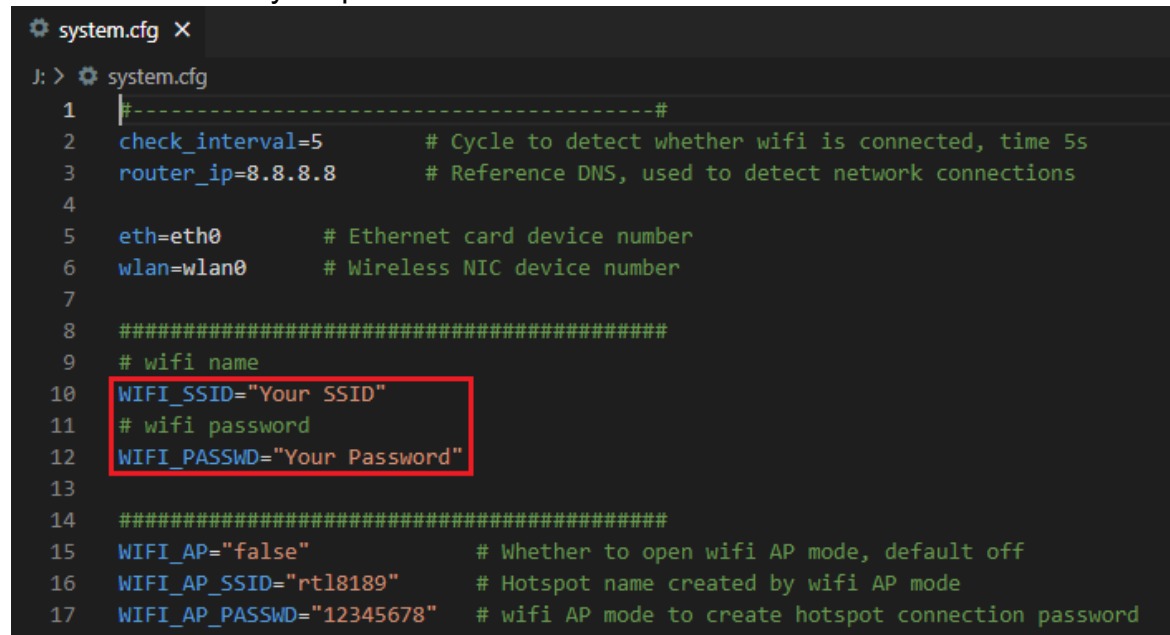
Note: Skip this step if you are using Ethernet port not using WiFi

After the OS image writing is completed, the microSD card will have a FAT32 recognized by the computer, find "system.cfg".



| 名称 | 修改日期 | 类型 | 大小 |
|-----------------------------|------------------|-----------------|-----------|
| dtb | 2022/11/9 2:50 | 文件夹 | |
| dtb-5.16.17-sun50iw9 | 2022/11/9 2:50 | 文件夹 | |
| gcode | 2022/11/9 10:35 | 文件夹 | |
| .next | 2022/11/9 2:50 | NEXT 文件 | 0 KB |
| BoardEnv.txt | 2022/11/9 2:53 | 文本文档 | 1 KB |
| boot.bmp | 2022/11/9 2:52 | BMP 图像 | 10 KB |
| boot.cmd | 2022/11/9 2:48 | Windows 命令脚本 | 4 KB |
| boot.scr | 2022/11/9 2:53 | 屏幕保护程序 | 4 KB |
| config-5.16.17-sun50iw9 | 2022/11/9 2:39 | 17-SUN50IW9 ... | 176 KB |
| Image | 2022/11/9 2:39 | 文件 | 20,631 KB |
| initrd.img-5.16.17-sun50iw9 | 2022/11/9 2:54 | 17-SUN50IW9 ... | 9,171 KB |
| system.cfg | 2022/11/10 17:52 | 文本文档 | 1 KB |
| System.map-5.16.17-sun50iw9 | 2022/11/9 2:39 | 17-SUN50IW9 ... | 4,239 KB |
| ulinitrd | 2022/11/9 2:54 | 文件 | 9,171 KB |
| vmlinuz-5.16.17-sun50iw9 | 2022/11/9 2:39 | 17-SUN50IW9 ... | 20,631 KB |

Open it with Notepad, replace WIFI-SSID with your WiFi name, and
PASSWORD with your password.

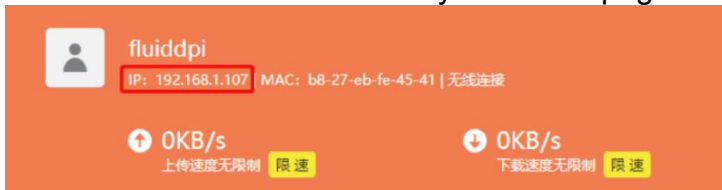


```
system.cfg X
J: > system.cfg
1 | #-----#
2 | check_interval=5      # Cycle to detect whether wifi is connected, time 5s
3 | router_ip=8.8.8.8    # Reference DNS, used to detect network connections
4 |
5 | eth=eth0             # Ethernet card device number
6 | wlan=wlan0          # Wireless NIC device number
7 |
8 | #####
9 | # wifi name
10 | WIFI_SSID="Your SSID"
11 | # wifi password
12 | WIFI_PASSWD="Your Password"
13 |
14 | #####
15 | WIFI_AP="false"     # Whether to open wifi AP mode, default off
16 | WIFI_AP_SSID="rtl8189" # Hotspot name created by wifi AP mode
17 | WIFI_AP_PASSWD="12345678" # wifi AP mode to create hotspot connection password
```

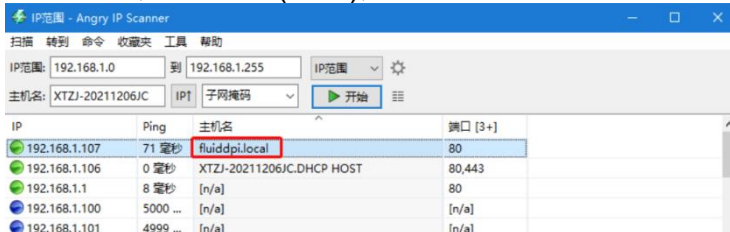
Configure the Motherboard

SSH Connect to Device

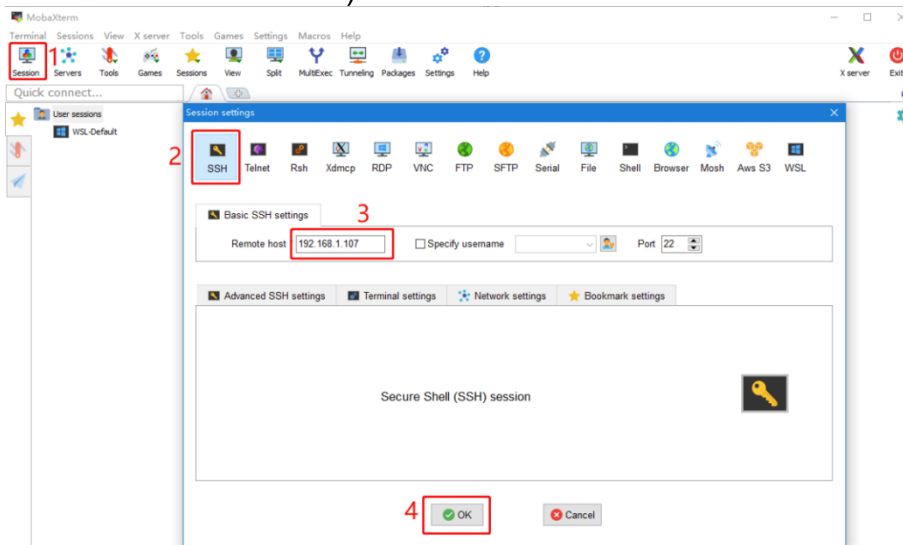
1. Install the SSH application Mobaxterm: <https://mobaxterm.mobatek.net/download-home-edition.html>
2. Insert micorSD card to SKRat, wait for system to load after power on, approx. 1-2min.
3. The device will automatically be assigned an IP address after being successfully connected to the network.
4. Find the device IP address on your router page.



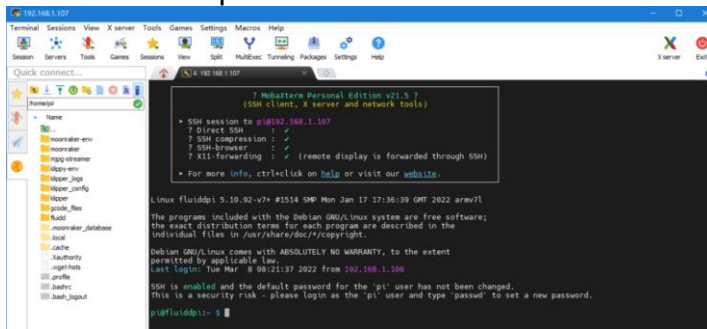
5. Or use the tool <https://angryip.org/>, scan all IP addresses in the current network organize by names, and find the IP named Fluidd, Mailsail(CM4), or BTT-CB1, Hurakan(CB1), as shown below.



6. Open Mobaxterm and click "Session", and click "SSH", inset the device IP into Remote host and click "OK" (note: your computer and the device need to be in the same network.)

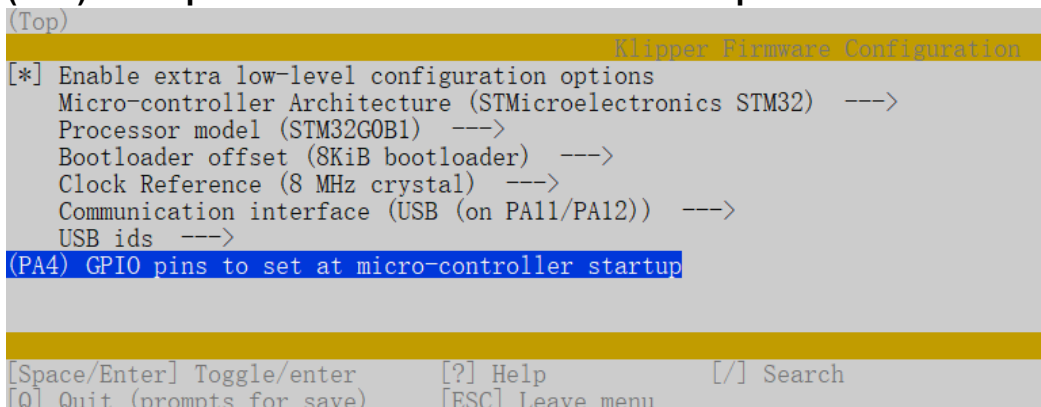


7. Login:
CM4:
Login as: pi
Password: raspberry
CB1:
Login as: biqu
Password: biqu

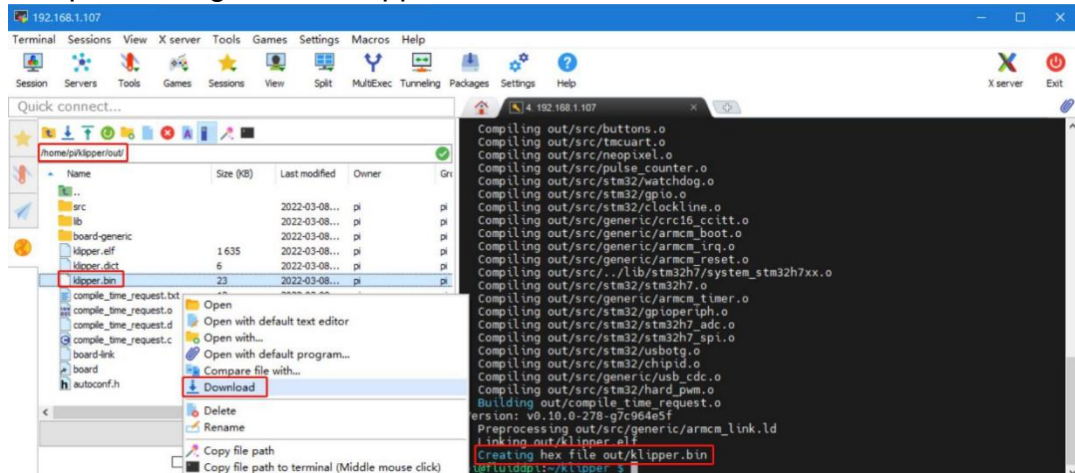


Compile MCU Firmware

- After SSH is successfully connected to the device, enter in terminal:
**cd ~/klipper/
make menuconfig**
Compile with the configuration shown below (if the options below are not available, please update your Klipper source code to the newest version)
*** [*] Enable extra low-level configuration options**
*** Micro-controller Architecture (STMicroelectronics STM32) --->**
*** Processor model (STM32G0B1) --->**
*** Bootloader offset (8KiB bootloader) --->**
*** Clock Reference (8 MHz crystal) --->**
When communicating via USB
*** Communication interface (USB (on PA11/PA12)) --->**
When communicating via CAN bus
*** Communication interface (CAN bus (on PD0/PD1)) --->**
(PA4) GPIO pins to set at micro-controller startup



2. Press 'q' to exit, and "Yes" when asked to save the configuration.
3. Run make to compile firmware, "klipper.bin" file will be generated in the home/pi/klipper/out folder when make is finished, download it onto your computer using the SSH application.



Firmware Update

Using microSD Card

1. Rename klipper.bin to "firmware.bin", copy it to the root directory of the microSD card, insert the microSD card into the card slot of the board, click the reset button, or power on again, the firmware will be updated automatically, after the update is complete, "firmware.bin" in the microSD card will be renamed to "FIRMWARE.CUR".
2. Enter `ls /dev/serial/by-id/` in the command line to check the motherboard ID to confirm whether the firmware is updated successfully as shown below.

```
pi@fluidpi:~/klipper $ ls /dev/serial/by-id/  
usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00  
pi@fluidpi:~/klipper $
```

copy and save this ID, it is needed when configuring the file.

Via DFU

If `ls /dev/serial/by-id/` can find the klipper device ID of the MCU, you can enter `make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00` directly to write the firmware. (Note: replace `/dev/serial/by-id/xxx` with the actual ID queried in the previous step.)

Shenzhen Big Tree Technology Co., Ltd.
BIGTREETECH

```
biqu@Hurakan:~/klipper$ make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
Building hid-flash
/bin/sh: 1: pkg-config: not found
hid-flash requires libusb-1.0, please install with:
sudo apt-get install libusb-1.0
Flashing out/klipper.bin to /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
Entering bootloader on /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
Device reconnect on /sys/devices/platform/soc/52000000.usb/usb1/1-1/1-1.1/1-1.1:1.0
sudo dfu-util -p 1-1.1 -R -a 0 -s 0x8002000:leave -D out/klipper.bin

dfu-util 0.9

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release!!!
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash "
Downloading to address = 0x08002000, size = 25264
Download [=====] 100% 25264 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status

Failed to flash to /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00: Error running dfu-util

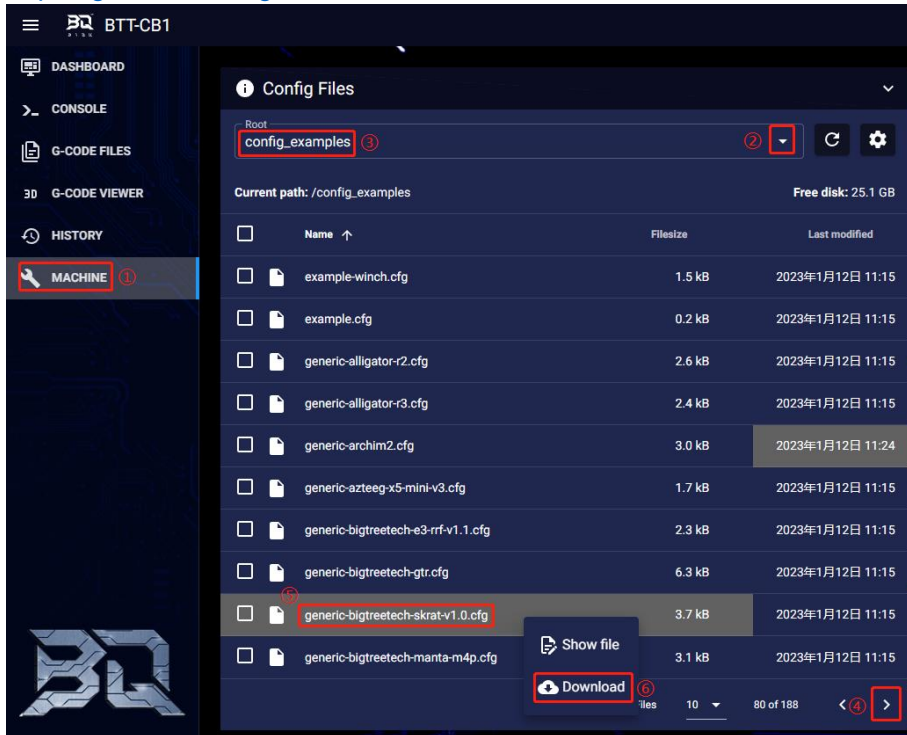
If the device is already in bootloader mode it can be flashed with the
following command:
make flash FLASH_DEVICE=0483:df11
OR
make flash FLASH_DEVICE=1209:beba

If attempting to flash via 3.3V serial, then use:
make serialflash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
```

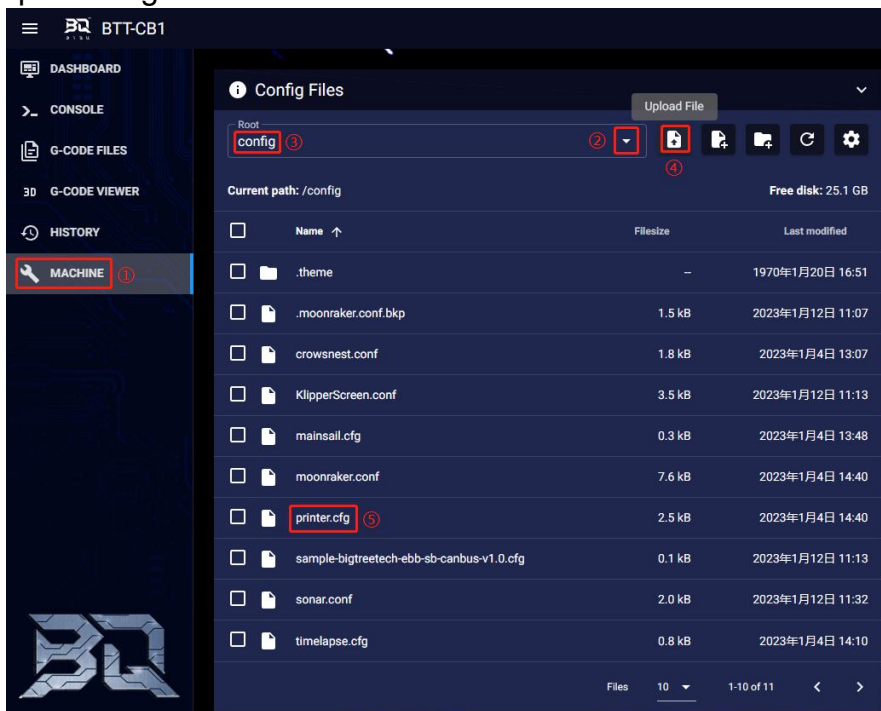
After the writing is completed, there will be an error message: dfu-util: Error during download get_status, just ignore it.

Configure Klipper

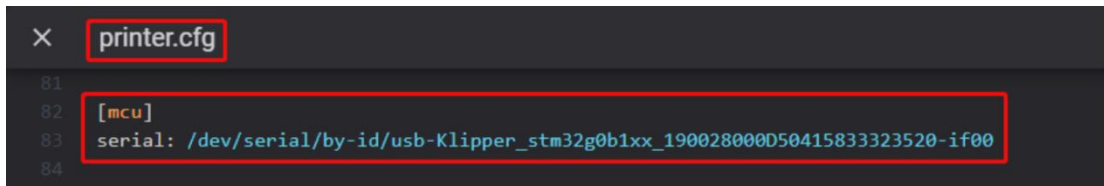
1. Enter your device IP address into your browser, and find the reference config for the motherboard in the directory shown below, if there is no such config available, update your Klipper source code to the newest version or download it from GitHub:
<https://github.com/bigtreotech/SKRat>



2. Upload your finished config file into Configuration Files, and rename it to "printer.cfg".



3. Enter the correct motherboard ID



```
× printer.cfg
81
82 [mcu]
83 serial: /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
84
```

Refer to <https://www.klipper3d.org/Overview.html> for detailed configuration guide according to your machine type.

Firmware Update

Update via microSD Card

1. Make sure the microSD card is formatted to FAT32.
2. Rename your firmware file to "firmware.bin" (note: make sure your system is showing file suffix, if the suffix is hided, "firmware.bin" will be shown as "firmware").
3. Copy "firmware.bin" to the root directory of your microSD card.
4. Insert microSD card to the motherboard and power on, the bootloader will automatically update the firmware.
5. The status indicator light will flash during the update process.
6. When the status indicator light stops flashing and the firmware.bin file has been renamed to firmware.cur, that is to say, the firmware has been successfully updated.

Cautions

1. Max. heated bed current is 10A, if high power heated bed is preferred, please use 24V to power the system and use a 24V heated bed.
2. To ensure proper operation of the CNC fan port, insert the voltage selection jumper.
3. The microSD card slot is not spring loaded, please be careful when inserting the microSD card to prevent damage to the card slot. BTT is not responsible for any damage caused by forcefully inserting the microSD card.

FAQ

Q: Max. current of the heated bed, heater cartridge, fan port?

A: Heated Bed: 10A Continuous, 15A Instantaneous
Heater Cartridge: 8A Continuous, 10A Instantaneous
Fan Port: 1A Continuous, 1.5A Instantaneous

The combined current of the driver, heater cartridge and fan port should not exceed 15A.

Q: Can not update the firmware with microSD card?

A: Make sure your microSD card has been formatted to FAT32, and the firmware file name is "firmware.bin", make sure your system is showing a file suffix, if the suffix is hided, "firmware.bin" will be shown as "firmware".